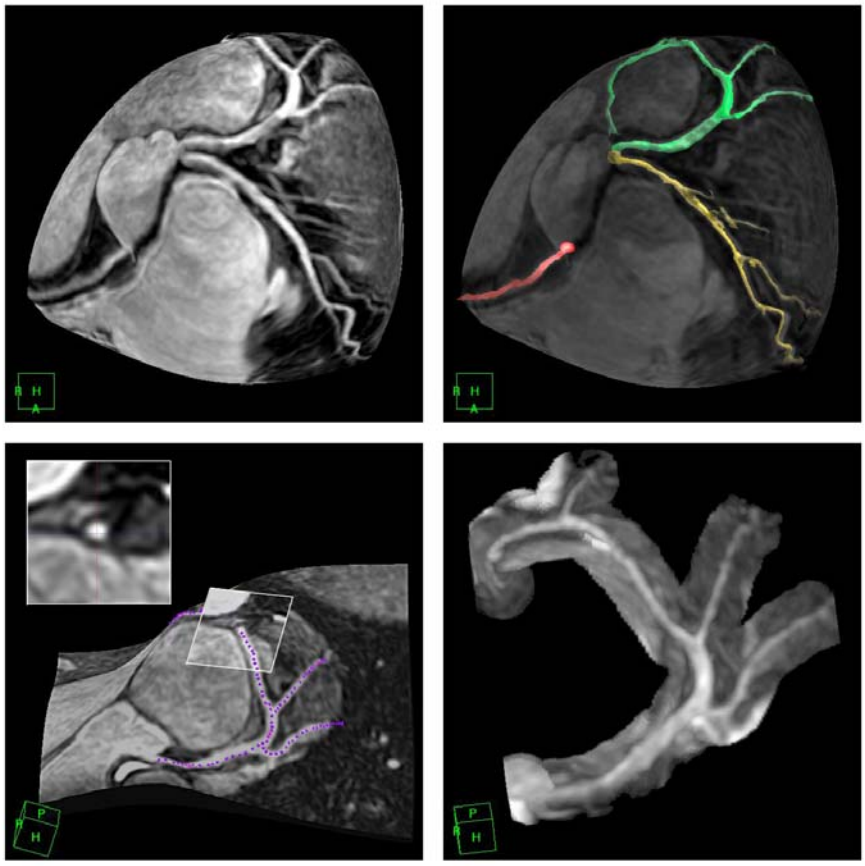
CoroViz: Visualization of 3D Whole-Heart Coronary Artery MRA Data

Master's Thesis

Stefan Tuchschmid



Supervision

Oliver M. Weber, PhD
Alastair J. Martin, PhD

Department of Radiology

University of California San Francisco
San Francisco, USA

Prof. Peter Bösiger, PhD

Institute for Biomedical Engineering

Swiss Federal Institute of Technology
Zurich, Switzerland

Start of Thesis: March 30, 2004

End of Thesis: September 30, 2004

1 Abstract

Whole-heart coronary MRA has been demonstrated to allow the imaging of the entire coronary tree in a single volume. However, a tool for comprehensive display of the 3D data sets has been lacking. To address this need, a software tool ('CoroViz') was developed. The advanced visualization modules depend on knowledge of the vessel centerline. Semi-automatic tracking based on at least one user defined point on the coronary tree was implemented, combining a vessel enhancement filter with a wave-front propagation algorithm. In all views, an interactive user interface allows zooming, panning and rotation of the data space along with a manual correction of the found centerline.

Globe visualization ('GlobeViz') utilizes a maximum intensity projection (MIP) onto a deformed sphere defined by the coronary vessels. The user can freely choose the thickness of the MIP-volume. Tube visualization ('TubeViz') creates a masked MIP, where the mask consists of a tube with user-defined diameter along the vessels. Globe and tube visualizations allow the three-dimensional assessment of the coronary arteries with only marginal concealment from signals originating outside the vessel. Besides the display of axial coronal and sagittal viewports, the current vessel cross-section is shown which permits the identification of vessel branching and stenosis. Additional modules include planar and curved reformats as well as a localized region-growing segmentation.

Quantification tools include interactive vessel length measurement and indication of maximum, minimum and average diameters along the vessel. First results and findings in healthy adults are presented. The use for clinical routine image post-processing and modalities other than coronary MRA remain to be investigated.

2 Preface

This work is part of my Master's thesis for the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland. I'd like to thank my corresponding Professor Dr. Peter Bösiger for the great opportunity to write this thesis here in San Francisco.

Oliver M. Weber, PhD, and Alastair J. Martin, PhD, have been the best support a graduating Master's student could wish for. None of this would have been possible without the creative input and strong encouragement I received from both of you.

Finally I thank the one person who has supported me through it all. 24'600 lines of code have been written, some of them in the middle of the night. It is done now, and the rewards have been higher than I ever expected. I dedicate this work to my love, Moni Oberholzer.

3 Contents

4 List of Figures

5 List of Tables

6 Introduction

In this thesis we present a software package called *CoroViz* for the visualization and quantification of three-dimensional coronary artery magnetic resonance angiographic data. The aim of this introduction is to provide the medical background needed to understand the motivation and requirements for the *CoroViz* project. The information is provided by the Texas Heart Institute (http://www.tmc.edu), the Surgical Associates of Texas (http://www.texheartsurgeons.com), and the American Heart Association (http://www.americanheart.org).

6.1 Coronary Anatomy

The coronary circulation system provides the heart with oxygen-rich blood of the aorta. Two main coronary blood vessels branch off, the *right coronary artery* (RCA) and the *left coronary artery*. The *left main* (LM) artery further branches into the *left anterior descending* (LAD) and the *left circumflex* (LCX) artery. These coronary arteries branch off into smaller arteries to supply oxygen-rich blood to the entire heart muscle. The right side of the heart is smaller and less muscular, responsible only for the blood transport to the lungs. The left side of the heart is bigger and pumps blood to other parts of the body. Figure 1 shows an overview of the coronary anatomy.

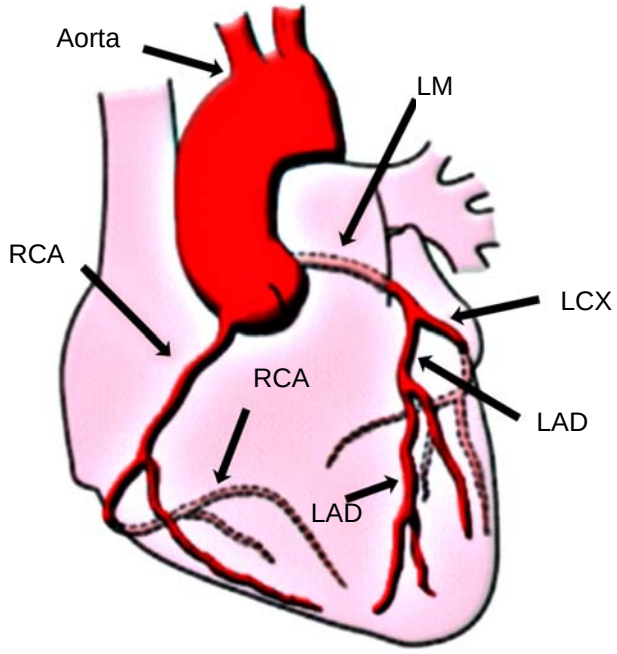


Figure 1: Coronary anatomy (adapted, © Texas Heart Institute)

6.2 Coronary Artery Disease

Coronary artery disease (CAD) is one of the major causes of morbidity and mortality in the western world. According to an estimate of the National Institute of Health, 7 million Americans are affected by the disease, and each year more than 500’000 men and women die of a heart attack caused by CAD. CAD often results from *atherosclerosis* (Greek: ‘athero’ = gruel or paste,

'scleroris' = hardness), a process in which deposits of fatty substances, cholesterol, cellular waste products, calcium and other substances build up in the inner lining of an artery (*plaques*). Figure 2 exhibits a cross-section of a diseased coronary artery.

Risk factors can be divided in uncontrollable (age, gender, heredity) and controllable. Controllable factors include high blood pressure, high blood cholesterol, smoking, obesity, physical inactivity, diabetes, and stress.

Atherosclerosis may be present without causing symptoms. Symptoms in the advanced stage include angina pectoris (Latin: 'strangling in the chest'), a sharp pain in the chest caused by the myocardial ischemia. Plaques can grow large enough to significantly reduce the blood's flow through an artery. However, most of the damage occurs when they become fragile and rupture, causing blood clots to form. If this process blocks a blood vessel (thrombosis), there is a sudden decrease in the blood flow which can cause a myocardial infarction.

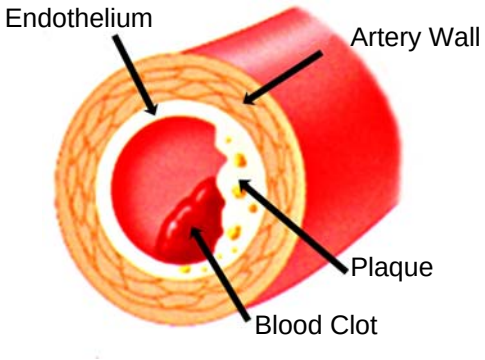


Figure 2: Cross-section of diseased coronary artery with plaques. (adapted, © Texas Heart Institute)

6.3 Diagnosis

Coronary artery disease is clinically diagnosed based on an assessment of patient symptoms, stethoscopy and a number of tests, including an *electrocardiogram* (ECG), exercise tests, *positron emission tomography* (PET) scanning, echocardiography and coronary angiography.

So far, *catheter x-ray angiography* is considered the primary diagnostic tool used for coronary angiography (Earls, Ho et al. 2002). X-Ray coronary angiography is performed in a cardiac catheterization laboratory. Cardiac catheterizations are used to define coronary anatomy and to guide patient therapy. However, x-ray coronary angiography is both expensive and invasive, with ionizing radiation exposure for the patient and the operator to be taken into account. Also, a small risk of serious complications exists. Thus, there is a strong need for a more cost-effective, non-invasive, and patient friendlier technique.

Coronary magnetic resonance angiography (MRA) combines several advantages and has great potential: due to its non-invasiveness it is patient friendly, a high spatial resolution can be obtained, it can gather 3 dimensional data, and there is no exposure to potentially harmful ionizing radiation. Also, by replacing several procedures with a 'one-stop shop', a MR based cardiac exam enables patients to get treatment sooner and at a lower cost than the traditional diagnostic pathway (Boer 2000).

However, the imaging of the coronary arteries by means of MRA is also very challenging: the coronaries are very small vessels (diameter of 2-4 mm) and have a high tortuosity. Moreover, they are constantly displaced by respiratory and cardiac motion. Substantial progress has been made in the last 20 years, with clinical studies demonstrating both the usefulness and the high imaging quality of the newer MR sequences used in coronary MRA. An overview of recent progress in cardiac MRI is found in (Earls, Ho et al. 2002).

At the University of San Francisco, a novel whole-heart coronary MRA sequence has been demonstrated to allow the imaging of the entire coronary tree in a single volume (Weber, Martin et al. 2003). Long segments of all major vessels can be captured with high quality and good discrimination from the background. The spatial resolution is high and almost isotropic, with a significantly reduced examination time compared to a double-oblique partial volume approach for all coronary vessels. The whole-heart coronary MRA sequence above is based on a steady-state free precession (SSFP) sequence (Deshpande, Shea et al. 2001; Shea, Deshpande et al. 2002; McCarthy, Shea et al. 2003).

Besides *coronary magnetic resonance angiography* (CA-MRA), *multi-detector array computed tomography* (MDCT) scanners have been used to create three-dimensional images of the coronaries minimal-invasively. The main advantages of MDCT scanning in respect to coronary imaging are a high contrast-to-noise ratio, little background signal, and a short examination time in the order of 30 seconds. Disadvantages include the inherent radiation exposure and the necessary use and application of an iodinated contrast agent. Moreover, current MDCT scanners rely on a heart rate smaller than about 70 beats per minutes, which often can only be obtained by sedating the patient.

It is not sure yet which method will become the coronary artery imaging method for the different patient groups and will eventually replace the invasive x-ray coronary angiography. However, the results for both MDCT and CMRA look very promising in the foreseeable future.

6.4 Treatment

Approaches for the treatment of coronary artery disease include medication, cardiovascular surgery and interventional cardiology. Even though the symptoms are reduced, the disease and its causes are still present after the treatment, demanding a modification of lifestyle in order to cure the disease.

Medication may include Nitroglycerin to dilate the arteries, Beta blockers to reduce the amount of oxygen the heart demands, and Aspirin to decrease the chances of a blood clot forming inside the artery. Cardiovascular surgery usually means bypass surgery. The blockage of the coronary artery is bypassed with a vessel (‘graft’) taken from another part of the patient’s body, usually the saphenous vein or the internal mammary artery. Figure 3 shows an example of graft placement, connecting the aorta with the left anterior descending.

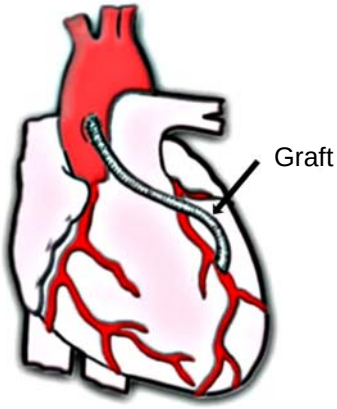


Figure 3: Coronary artery bypass surgery (adapted, © Texas Heart Institute)

In interventional cardiology, a catheter guided tool is used to physically open up a blockage. The catheter is usually inserted into an artery in the leg. For angioplasty, the catheter has a small balloon on its tip, allowing a flattening and partial removal of the plaque by inflation of the balloon. Sometimes, a stent procedure is used along balloon angioplasty. The stent is placed after the inflation of the balloon, effectively keeping the artery open and stopping the artery from collapsing. Further interventional methods include atherectomy (a drill is used to shave plaque from wall), or laser ablation. The large number of possible treatments requires a reliable, safe and trustworthy assessment of the current condition.

7 Motivation

The three-dimensional nature of MRI allows for very sophisticated visualization of the vascular system, such as volume rendering or maximum intensity projections from various directions. Also, the operator is able to rotate, pan and zoom the data in all directions, providing the best possible view position. Although there is a large number of post-processing software packages available from manufactures of radiological workstations, the use for the adequate visualization and quantification of CA-MRA has been limited. This section intends to illuminate the challenges met in the visualization process and tries to explain why and where general-purpose software packages fail. An overview of current visualization and analysis methodologies is found in (Udupa 1999).

The goal of this work is to develop a comprehensive tool for 2D and 3D visualization and quantification of whole-heart coronary MRA data as provided by (Weber, Martin et al. 2003). The method provides high-resolution, three-dimensional data of the heart in unprecedented quality. However, the adequate visualization of the coronary arteries from MRA data remains very challenging.

Firstly, the diameter of the vessels is often less than 1 mm, meaning a cross section size of maybe 4 voxels. Even though the data might be interpolated for clinical examination, the inherent information content remains the same. Figure 4 shows sample cross section images of the proximal and distal part of the left circumflex coronary artery.

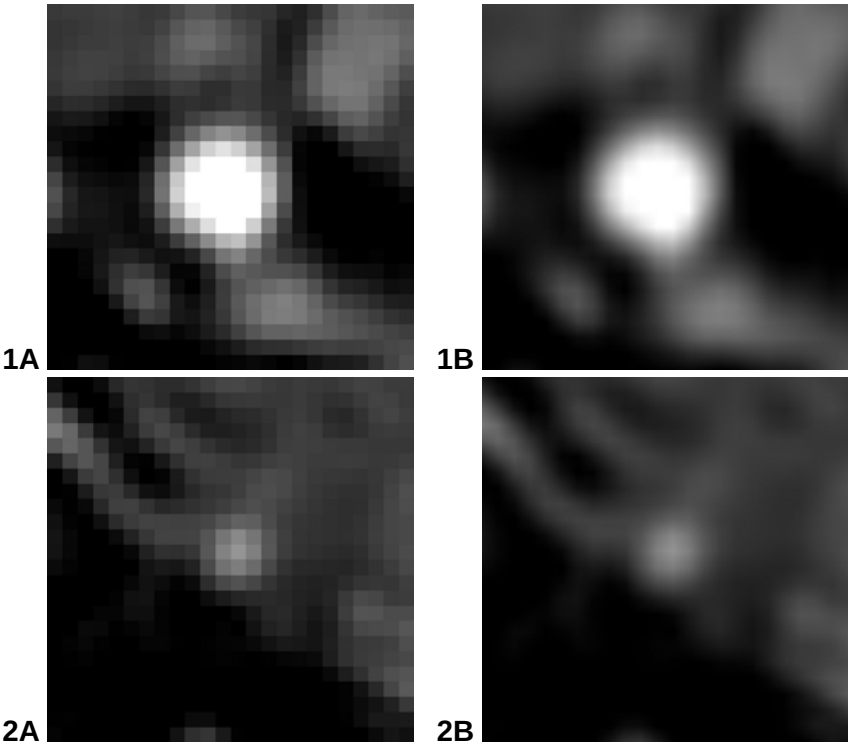


Figure 4: Cross section images of the proximal (1A, 1B) and distal (2A, 2B) part of the LCX. The inherent information content is the same for the interpolated images (1B, 2B) and the original data (1A, 2A).

Secondly, large signal contributions originate from outside the vessel compared to image data gathered from x-ray angiography or MDCT where the use of contrast fluids greatly enhances the vessel-to-background distinction. The small and tortuous vessels, combined with massive signals from organs beside the coronary arteries make an automatic extraction of the centerline and/or an automatic segmentation very challenging.

7.1 Current approaches

Unlike the targeted-volume technique, the whole-heart sequence is not aimed at a certain vessel, e.g. the right coronary artery. The traditional targeted approach requires setting up the double-oblique sub volume on the scanner prior to collecting the data. This step assures that the vessel to be investigated lies mainly in plane. The whole-heart approach eliminates the need for a survey scan showing the coronaries and a subsequent planning of the imaging volume. However, the vessels are no longer aligned along the imaging planes.

The current approach for post-processing of whole-heart MRA data is to create multi-planar reformatted (MPR) slices in analogy to the targeting technique used in the acquisition planning. Figure 5 shows an example of multi-planar reformatting for different vessel combinations. The result of the MPR step is an interpolated image stack with the wanted vessels lying in-plane as optimal as possible.

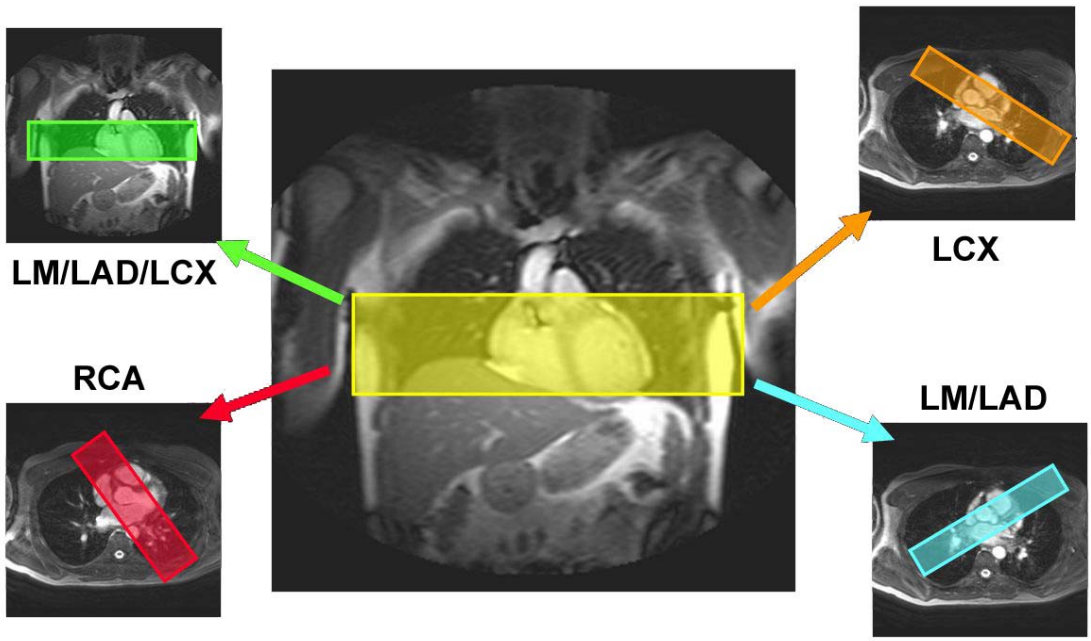


Figure 5: Conventional data aquisition is targeted at different vessels.The whole-heart approach (middle) allows an imaging of the whole coronary system. However, multi-planar reformats are required to ensure that the wanted vessels are aligned in the imaging planes (left and right side).

However, even though the slices in the image stack are now aligned towards the vessel under investigation, further steps have to be taken to allow the visualization of the whole vessel in one single image. The current reformatting tool on the standard Philips image post processing

station (EasyVision, Philips, Best, The Netherlands) produces a single image (*curved reformat*) showing one entire coronary artery out of a 3D data set. Navigating through the data, the operator defines a series of points on a coronary artery. The tool then produces a single image of the coronary artery, based on the selected points. The resulting reformatted image can be saved. Subsequently, the length of the vessel can be measured on the reformatted image.

For research purposes, a more robust and intuitive tool called Soap-Bubble has been introduced (Etienne, Botnar et al. 2002). The current implementation is intended for targeted volume data only and doesn't take advantage of the almost isotropic resolution acquired by the Whole-Heart approach. Figure 6 shows Soap-Bubble reformatted images of MRA data. The main disadvantage of the Soap-Bubble tool in respect to whole-heart data is that MPR's for the different vessels have to be made prior to Soap-Bubble processing. Moreover, three-dimensional information is lost and the Soap-Bubble images don't preserve distances in the output images.

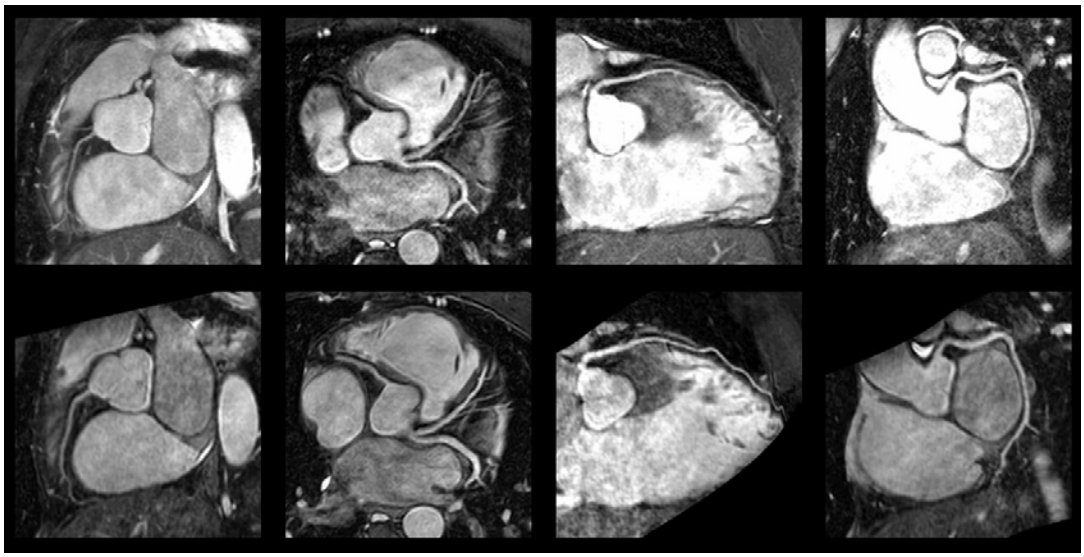


Figure 6: Soap-Bubble reformatted images. Source data for the top row was obtained with targeted acquisition, for the bottom row with the whole-heart sequence.

Alternatively, a volume rendering approach has recently been reported to be useful in coronary MRA display (Yasutaka Ichikawa and Makino 2004). Volume rendering software such as VirtualPlace Advance (AZE Ltd, Tokyo, Japan) use the three-dimensional data to create an interactive virtual scene. In a first step, the operator coarsely segments the data by outlining the heart in several tomographic slices. The contour ('mask') is then interpolated and the heart cut out. A manual refinement is needed in most cases. The operator manually adjusts the mask in either the two-dimensional view ports or on the three-dimensionally rendered heart, removing tissues that block the view to the arteries.

The heart is rendered by tracing rays through the masked volume. The image intensity at a given point within the volume defines the color and transparency of that point as defined in separate color/transparency lookup tables. The rays are then attenuated and colored depending on their route through the volume. Figure 7 depicts a rendered heart from two different view angles.

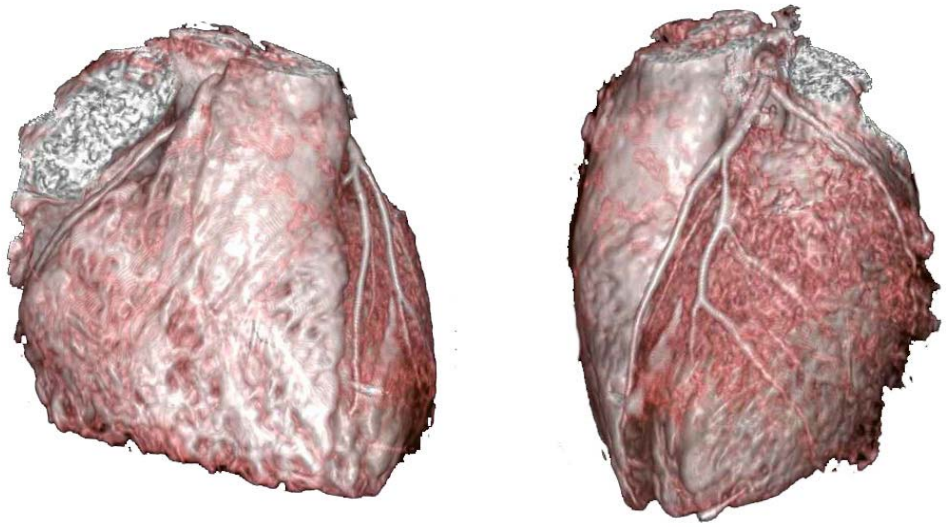


Figure 7: Volume rendered heart showing the coronary arteries from two different view angles. The rendering has been done with the VirtualPlace Advance software (Aze Ltd., Tokyo, Japan).

7.2 Limitations of the current approaches

To further motivate the CoroViz project, the main shortcomings of the Soap-Bubble tool as well as limitations of the general volume rendering approach are presented here. A comparison in respect to the length of the visualized arteries can be found in (Yasutaka Ichikawa and Makino 2004).

7.2.1 Limitations of the Soap-Bubble tool

The original idea of the Soap-Bubble tool (Etienne 2001; Etienne, Botnar et al. 2002) is based on the assumption that the coronary anatomy fits to a relatively smooth three-dimensional surface. This surface ('Soap-Bubble') is then deformed according to user defined points on the coronary tree. Finally, a parallel MIP to a plane parallel to the slice direction yields the two-dimensional reformatted image. Figure 8 depicts the idea behind the Soap-Bubble coronary re-formatting.

Whereas the Soap-Bubble tool provides adequate visualization of targeted data with vessel braches lying mainly in-plane, the projection is problematical for parts of the vessel that travel steeply in z-direction. The reformatted image is made from a projection in the z-axis and therefore the vessel appears shorter than it actually is. Because it is not possible to fit the whole coronary artery in a thin volume, the use of the Soap-Bubble tool for whole-heart MRA data is limited due to the artificially shortened vessel branches. Moreover, image resolution for perpendicular vessels is poor since the interpolation grid is equally spaced in the xy-plane. Figure 9 shows the application of the Soap-Bubble tool for whole-heart MRA data.

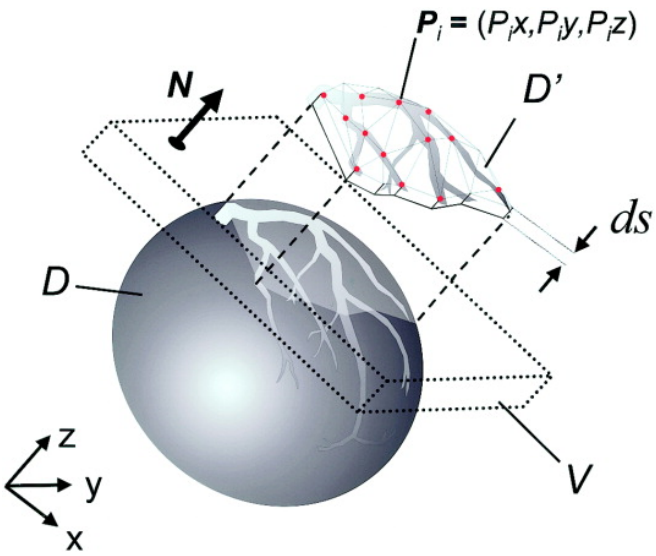


Figure 8: Soap-Bubble coronary reformatting. '3D coronary MRA data are acquired in a Cartesian coordinate system (x,y,z) and in the volume V. The user-identified points Pi define the manipulated surface D, which is shown in a close-up view as a curved (3D Delaunay triangulation), convex hull D'. A parallel MIP of this manipulated surface (parallel to the slice-selection direction N) leads to the planar coronary reformat. MIP can selectively be performed in a volume that closely encompasses the coronary arterial tree and is specified by the "skin thickness" ds.' Figure and quotation from (Etienne, Botnar et al. 2002).

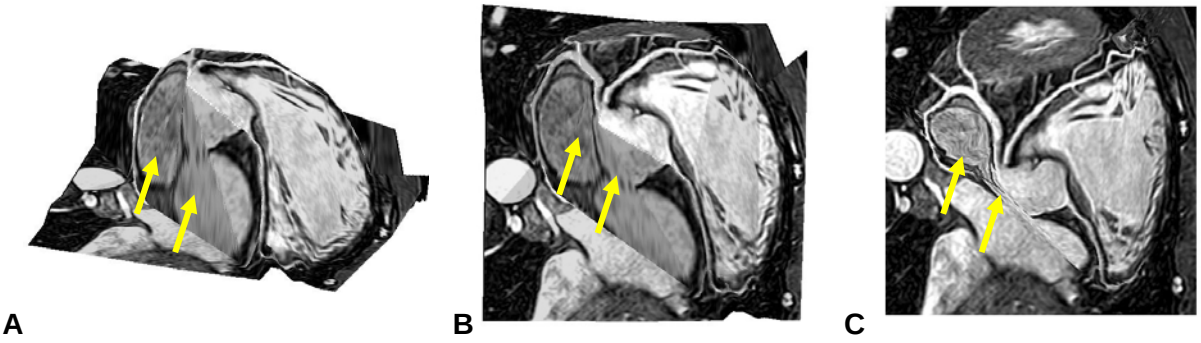


Figure 9: Soap-Bubble reformatting fails to accurately visualize distal branches of the coronary tree. (A,B) show the manipulated surface prior to the plane projection from different view angles. (C) depicts the aquired reformat. The distortions are best visible in areas where the surface is almost parallel to the projection direction (see arrows).

7.2.2 Limitations of Volume Rendering Tools

While the results of volume rendering with the VirtualPlace Advance program look sophisticated, the visualization heavily depends on the choice of the lookup tables as well as on the experience of the operator. Cutting out the heart can be a tricky process, especially if the arteries are embedded in surrounding tissues with similar intensity levels. Even more restricting for clinical use is the fact that the choice of contrast influences the thickness of the visualized coronary arteries. It is possible to artificially generate and obliterate stenosis simply by changing the contrast window and level. Figure 10 shows an example.

There is a risk that the human operator feels over-confident in the assessment of the data because the rendered scenes look very natural. It's important to realize that the underlying data is a one-dimensional intensity value and has a rather coarse resolution. Approaches closer to the source data (e.g. maximum intensity projections) may be preferable for the clinical evaluation of pathological cases.

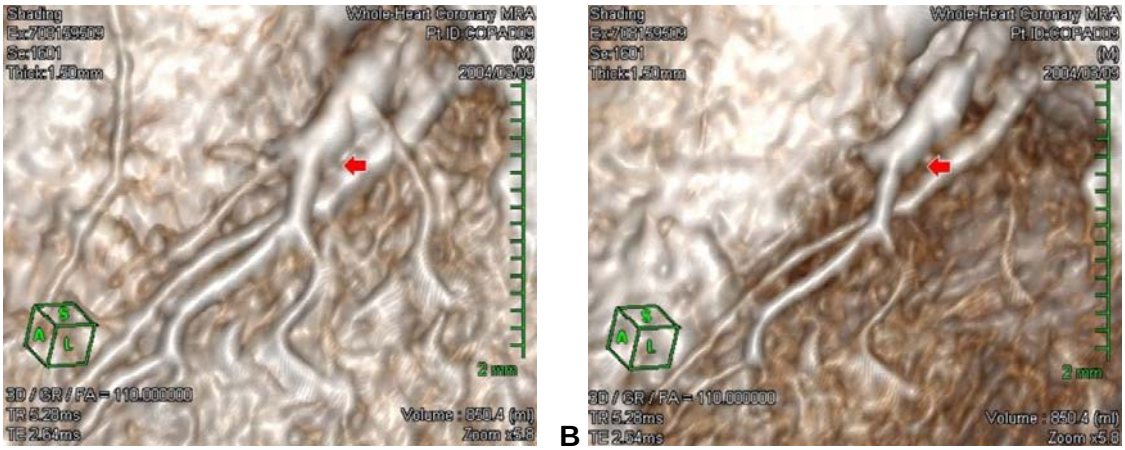


Figure 10: Volume rendering may artificially introduce or obliterate stenoses. (A,B) show the volume rendering of the same data with a different contrast setting. In (A) a number of side branches are visible that dissappear in (B).

7.3 Requirements

The aim of the CoroViz project is to develop a comprehensive tool for 2D and 3D visualization and quantification of whole-heart coronary MRA data sets.

The main design principle is the restriction to visualization techniques that stay close to the original source data. All visualization steps should be transparent to the human operator. Maximum intensity projections (MIPs) on virtual spheres, ellipsoids, and curved-fitted surfaces are explored. We investigate the possibility of automatic vessel tracking as well as vessel segmentation. The implemented quantification tool should be compared to the results acquired with the Soap-Bubble Tool and VirtualPlace Advance software.

The visualization tool is implemented on IDL and on a PC platform under WindowsXP. Real patient data is used during all phases of software development.

8 Methods

This section gives an overview of the CoroViz program and describes methods and approaches used for the different visualization modules. The actual implementation is described in the appendix and of course in the source code written for the CoroViz project. However, an indication of the general implementation strategy is included whenever appropriate.

8.1 The CoroViz Program

The development of the CoroViz software was the main task of this Master's thesis. It was implemented under IDL 6.1 (Interactive Data Language; Research Systems, Inc; Boulder, CO) on a commercial Windows XP (Microsoft Corp.; Redmond, WA) desktop computer with a 3.0 GHz Pentium 4 Processor (Dell Dimension 8300, Dell Computers, Austin, TX). A user manual and an installation guide are appended.

The main design principle is a restriction to visualization ideas that are transparent to the human operator. All the a-priori knowledge used while creating the visualizations is disclosed. In addition, the different visualization modules are depicted in the same three-dimensional scene, allowing a comparison and assessment of the spatial composition. The scene can be interactively zoomed, rotated, panned, and windowed.

The look-and-feel is made similar to standard software packages such as Gyroview or EasyVison, therefore users with general scanner software experience will feel familiar with the graphical user interface. Figure 11 depicts the workflow in the CoroViz program. Underlying ideas and implementation strategies are further described in the corresponding chapters.

The currently implemented modules include:

- Planar, based on Chapter 8.3: Multi-Planar Reformat (MPR)
- Curved, based on Chapter 8.4: Curved Reformat
- Globe Viz, based on Chapter 8.8: Globe Visualization
- Segmentation, based on Chapter 8.7: Vessel Segmentation and Surface Creation
- Filtering, based on Chapter 8.5: Hessian-Matrix based Filtering
- Measurements, based on Chapter 8.10: Quantification and Measurements
- Volume Viz, based on Chapter 8.9: Masked Volume Projection
- Tracking, based on Chapter 8.6: Vessel Tracking

The visualization modules depend on the knowledge of the vessel axis or centerline. Points belonging to the coronal tree can be either manually defined or are supplied by the semi-automatic tracking algorithm. Definition, refinement, and manual correction of vessel points are done interactively with mouse clicks on either the two- or three-dimensional viewports.

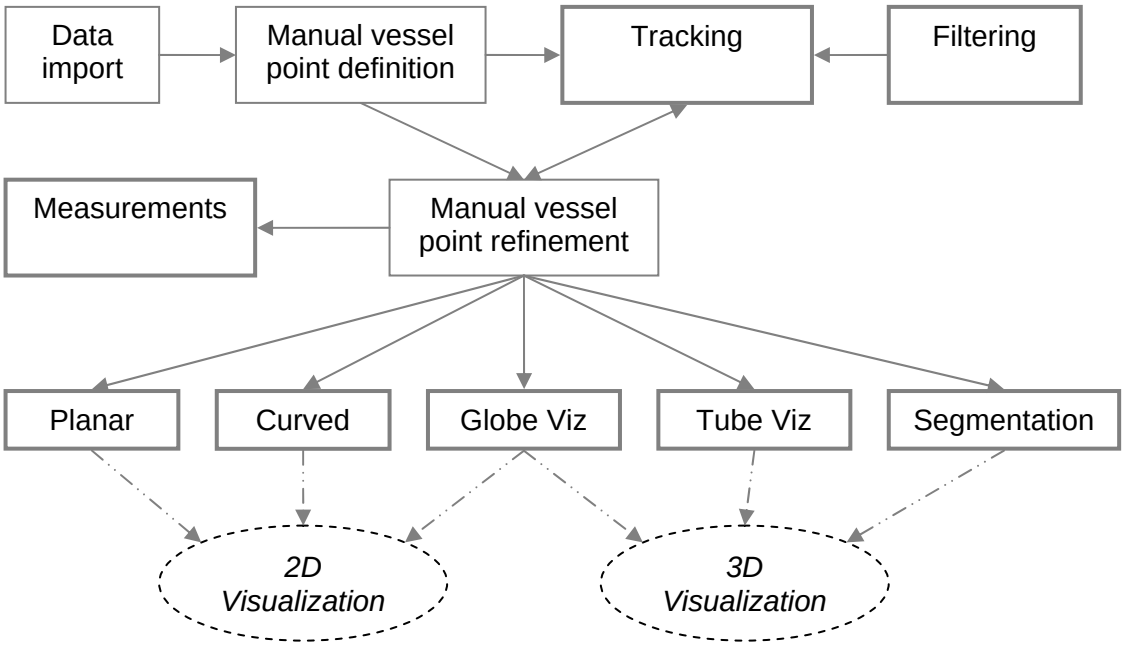


Figure 11: Workflow in the CoroViz program. The modules shown with solid-thick boxes are described in separate sections.

8.2 Input

Development and testing of the CoroViz software has been done with real-world patient data. Patients were scanned in a supine position (1.5 T Intera I/T, Philips Medical Systems, Best, The Netherlands). The three dimensional image data used for the visualization of the coronary anatomy is provided by the previously described whole-heart steady-state MRA sequence (Weber, Martin et al. 2003). The entire heart is covered in 140-160 slices. The slice thickness is 1.5 mm with an overlap of 0.75 mm, and is reconstructed to slices with 0.75 mm thickness. The in-plane image with a resolution of 1 x 1 mm^2 (Field-of-View 256 mm) is resampled to a 512 x 512 matrix. Therefore, the resulting voxel size is 0.5 x 0.5 x 0.75 mm^3, and the covered volume for a 140 slice set amounts to 256 x 256 x 105 mm^3.

8.3 Multi-Planar Reformat (MPR)

The aim of the multi-planar reformatting step is (a) to fit a plane through the points defined on the vessel, and (b) to move the plane perpendicular to the plane normal. This step is conceptually similar to rotating and moving the volume so the vessel is centered and lies in plane with the volume slices.

The minimum number of points needed to define a plane is three. If there are more points defined on the vessel, the plane is fitted in a way that the sum of the squared distances from the points to the plane is minimized. Figure 12 shows how the plane is fitted to detected points on the coronary vessel.

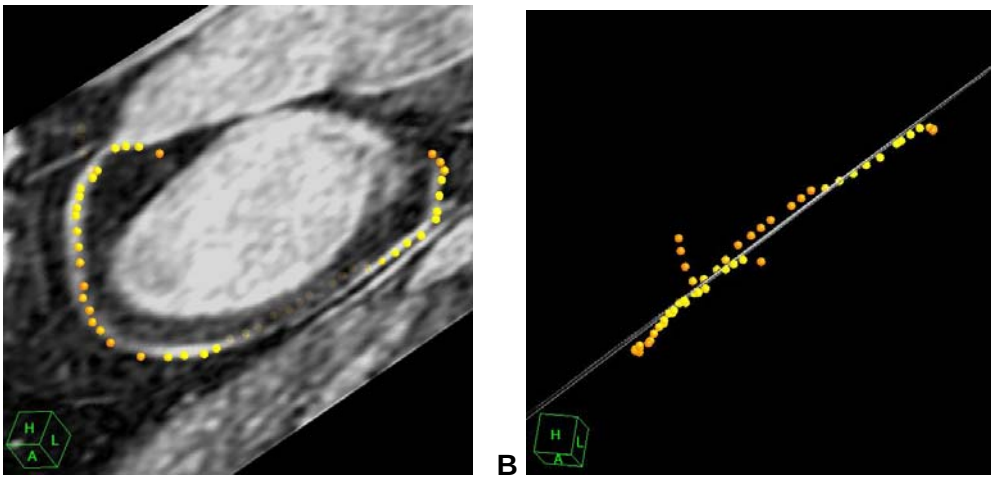


Figure 12: The plane fitted to points selected on the right coronary artery (A). The sum of the squared distances from the points to the plane is minimized (B).

8.3.1 Implementation

In a first step, a plane fitting algorithm is employed to find the position and direction of the center plane. Secondly, the reformatted image stack is generated by moving the center plane in parallel to the plane normal and sampling the original data at the given positions.

8.3.1.1 *Plane Fitting*

The algorithm used for minimizing the squared distances from points to plane is described in the Appendix. It is shown that the problem can be reduced to the solution of an eigenvalues equation system, where the eigenvectors are perpendicular to each other and describe the best, and intermediate and worst direction for the plane normal.

8.3.1.2 *Slice projection*

The center plane is defined by the center position and a plane normal as provided by the plane fitting algorithm. The specified planar slice is then extracted from the volumetric data using trilinear interpolation. Depending on the required number of slices, plane thickness, and plane

gap, an image stack is generated with the images having varying offsets in the direction of the plane normal. All the slices that are part of the projection volume for the given image are then projected onto the center plane.

8.3.2 Options for the Multi-Planar Reformat

Options for the Reformat-Module include the specification of a center offset, the slice thickness and slice gap as well as a choice of the projection modality used. All the options can be interactively set in the CoroViz program.

8.3.2.1 *Center offset*

Whereas the direction of the center plane is defined by the plane-fitting algorithm, the plane can be interactively moved in direction of the plane normal, allowing adjacent segments of the vessel to be visualized. Figure 13 shows an example of the RCA with two different offsets.

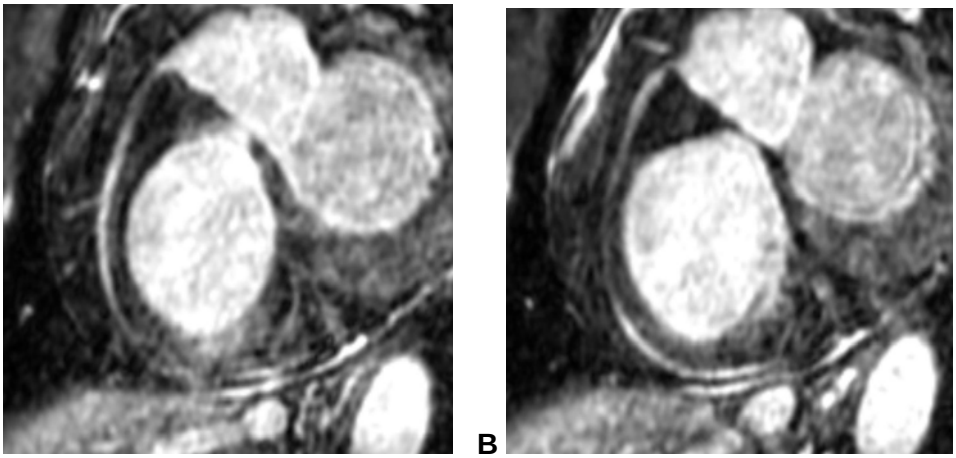


Figure 13: Two individual slices of the multiplanar reformat, showing adjacent segments of the RCA with offsets -1 mm (A) and +1 mm (B).

8.3.2.2 *Slice thickness and slice gap*

The slice thickness defines the width of the projection volume in direction of the plane normal. The distance between adjacent projection volumes is defined as the slice gap, with a negative gap indicating an overlap of the projection volumes.

8.3.2.3 *Projection type*

Types of projections include maximum (MIP), minimum (MinIP) and average intensity. In MRA, mostly MIP projections are used, allowing the display of vessel segments that are covered in different images of the reformatted stack. The choice of the slice thickness is crucial for a successful display of all the segments. If the selected slice thickness is too small (Figure 14-A), parts of the vessel are not covered by the projection volume, introducing image artifacts similar to a

stenosis of the vessel. The projection of a thick slice will fully cover the vessel, yet might block the view with signals from anterior or posterior of the vessel (Figure 14-D).

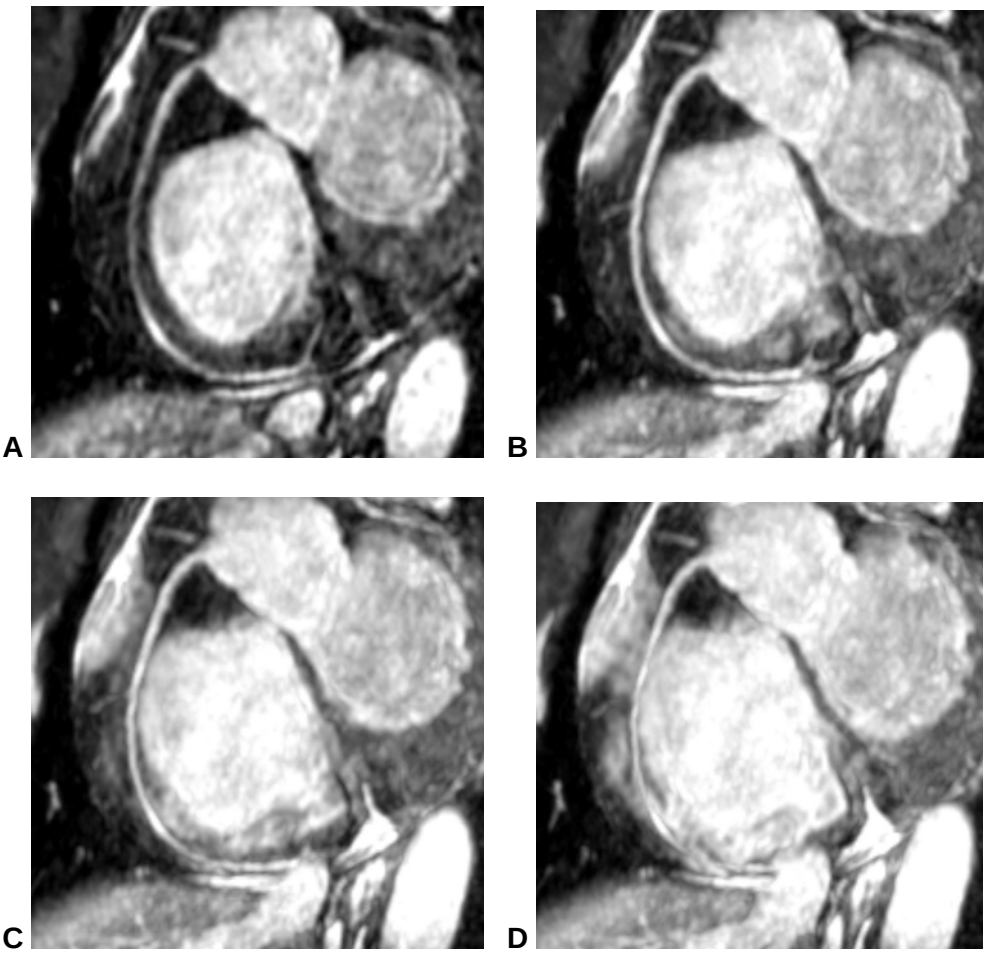


Figure 14: Maximum intensity projections of the multiplanar reformat, with 1 mm (A), 4 mm (B), 6 mm (C), and 8 mm (D) thickness.

8.4 Curved Reformat

The goal of curved reformation is to make a vessel visible in its entire length in one image. A MIP over a multi-planar reformatted stack succeeds in displaying vessels that are neither too tortuous nor hidden by surrounding tissues. In general, MIP's of the RCA provide good vessel depiction, whereas MIP's of the left system fail to provide adequate visualization. Figure 15 shows an example.

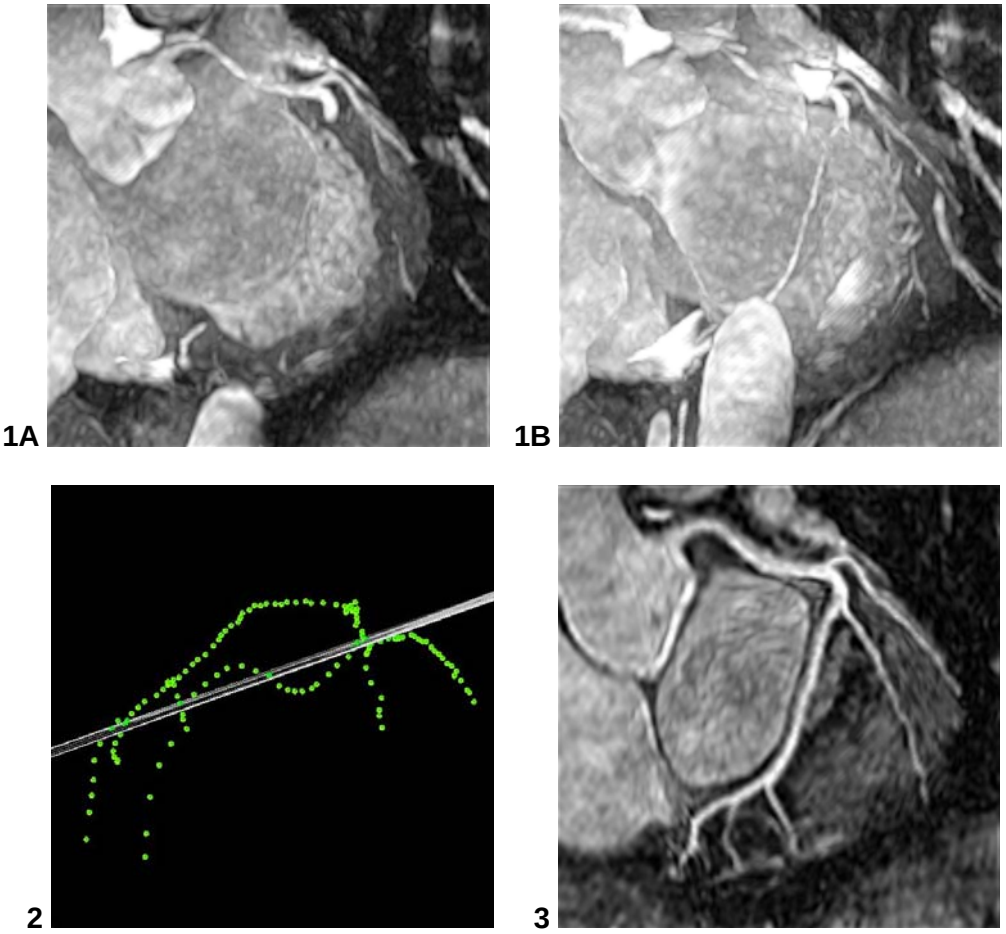


Figure 15: Maximum intensity projections of straight planes with 10 mm (1A) und 30 mm (1B) thickness fail to provide adequate visualization of the LCX due to bright surrounding tissues and a vessel that does not lay in plane with the MIP volume (see 2, showing a perpendicular view on the center-fitted plane together with the points defined on the vessel). More advanced techniqes such as a curved reformat (3) provide better visualization of the coronary tree.

Another way to display tubular structures such as the coronary arteries is to generate longitudinal cross-sections in order to show their lumen and the surrounding tissue in a curved plane. This process aligns the reformatted images with the orthogonal projections of a straight line or curved path. This procedure is also known as *curved planar reformation* (CPR) or curved multi-planar reformat.

Different options to generate a curved reformat with a-priori knowledge of the vessel center-line are addressed in (Kanitsar, Fleischmann et al.). However, *stretched*, *straightened*, and *projected* CPR all depend on a highly accurate vessel center-line. Moreover, only one tubular structure of the entire vascular tree can be visualized at the same time. The determination of the topographic position is difficult due to the distortion of the surrounding tissues, especially in the case of the straightened CPR where only short segments of visible side branches give an indication on the whereabouts.

To be able to visualize branching vessels as well, we rely on the idea of the *Soap-Bubble* tool. The original idea was to deform an ellipsoid or a sphere locally based on user-defined points on the coronary anatomy. The flattened surface of the bubble would have been the reformatted image. (Etienne 2001; Etienne, Botnar et al. 2002). However, at the time when the Soap-Bubble tool was coded, the predominant sequences of MRA all targeted a double-oblique volume, with most of the coronary anatomy lying in plane. Moreover, the targeted volumes were different for the left and the right coronaries. The current implementation of the Soap-Bubble tool therefore deforms a simple plane instead of an ellipsoid. For clarity I will refer to this idea as *Planar Soap-Bubble*, as opposed to the original Sphere Soap-Bubble idea.

Whole-Heart sequences provide data of the entire coronary tree, including proximal and distal parts of the RCA, the LM, the LCX and the LAD. With the Multi-Planar Reformat we are able to automatically extract a sub-volume of the data similar to the double-oblique targeted approaches. Therefore, the original in-plane assumption of the Planar Soap-Bubble holds true again. The implementation of the Curved Reformat tool basically relies on the Planar Soap-Bubble idea, with the difference that the data input is not the original volume, but the output of the *multi-planar reformat* procedure. Conceptually, the process can be broken down into 2 steps: A) Surface Creation, and B) Reformat Projection.

8.4.1 Surface Creation

The coronary vessel is identified by a set of points $P_i = \{X_i, Y_i, Z_i\}$ belonging to the arterial tree. The defined point coordinates correspond to the original data. Therefore, the coordinates have to be transformed first in to the MPR coordinate system with points $P_i' = \{X_i', Y_i', Z_i'\}$.

All combinations of affine transformation such as translation and rotation can be combined into a 4 by 4 matrix M with

$$\begin{bmatrix} X_i' \\ Y_i' \\ Z_i' \\ 1 \end{bmatrix} = M \circ \begin{bmatrix} X_i \\ Y_i \\ Zi \\ 1 \end{bmatrix}$$

The matrix *M* is defined during the plane-fitting process and depends on the center and normal of the reformatted center plane.

Smooth surface creation corresponds to the interpolation of an irregularly-gridded set of points with a surface. This process is also used e.g. in geodesy as an intermediate step for the construction of a digital elevation model. The object is to find a surface for the wanted area that is both smooth and intersects the points $\{X_i', Y_i'\}$ at elevation Z_i. First, a Delaunay triangulation of a

planar set of points is created, only taking the X- and Y-coordinates of the points into account. Once the triangles are defined, the Z-coordinates of the points are taken back into account and the triangles are hung in a 3D space, creating a two-dimensional surface.

Once all the triangles are defined, the elevation coordinates K_z may be interpolated for the desired grid coordinates $\{x, y\}$ using linear, quintic, minimum curvature or thin-plate spline interpolation. The values of $K_z(x, y)$ depict the z Position as a function of the x and y coordinates of the grid array.

8.4.2 Implementation

The TRI_SURF function interpolates a regularly- or irregularly-gridded set of points with a smooth quintic surface. TRI_SURF is similar to MIN_CURVE_SURF but the surface fitted is a smooth surface, not a minimum curvature surface. Internally, TRI_SURF calls first the TRIANGULATE procedure, which constructs a Delaunay triangulation of a planar set of points. Since Delaunay triangulations have the property that the circumcircle of any triangle in the triangulation contains no other vertices in its interior, interpolated values are only computed from nearby points. Eventually, TRIGRID is invoked to interpolate surface values to a regular grid.

Even though TRI_SURF returns a smooth surface, the quintic interpolation has proven to give unsatisfactory results for the curved reformatting process and often fails completely. By applying the keyword /LINEAR, the interpolation of the hung triangles is done in a linear mode. The linear interpolation provides a very robust surface generation method. However, since the first partial derivatives are not continuous, the results often look edgy, even on the projected image.

GRID_TPS uses thin plate splines to interpolate the set of values to the regular two dimensional grid. Thin plate splines are ideal for modeling functions with local distortions. The generated surface is smooth with continuous first partial derivates and passes through the original points $P_i' = \{X_i', Y_i', Z_i'\}$, therefore providing an almost ideal surface for data sampling and projection. However, the computation of the thin plate splines involves a Cholesky factorization which fails if the input points are very close together and have different elevation values $K_z(x, y)$, resulting in high values for the partial derivatives. Also, at least 7 non-linear points have to be defined on the vessel.

The current implementation uses the GRID_TPS surface generation function whenever possible. If the creation of the surface fails for whatever reasons, the GRID_TPS returns a scalar result and the operator is informed that the procedure failed. In this case the more stable TRI_SURF with the /LINEAR keyword set is used. Figure 16 shows a comparison of the curved images generated with both procedures.

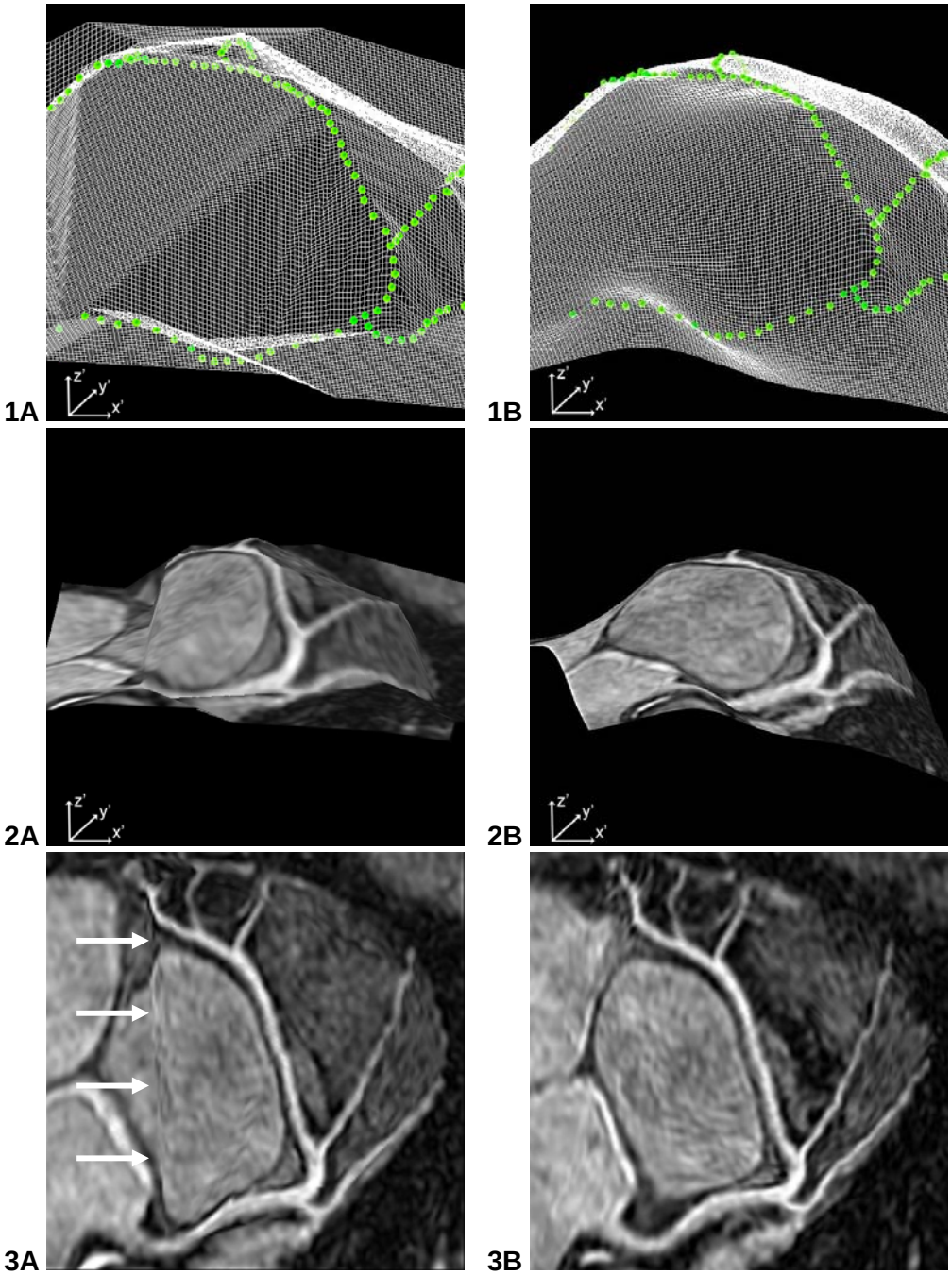


Figure 16: Triangulation and curved surface creation. Left side (A): Linear interpolation using the TRI_SURF function. Right side (B): Thin plate spline interpolation using GRID_TPS. The input points are triangulated, and the surface is interpolated, resulting in a mesh grid (1A,1B). Secondly, the data is sampled at the mesh points (2A,2B). The sampled data is then projected onto the x',y' plane (3A,3B). Whereas the thin plate method results in a smooth curved reformat, the linear interpolation leads to image artefacts (see arrows).

8.4.3 Reformat Projection

The interpolated points $\{x, y, K_z(x, y)\}$ define a surface through the reformatted volume. The data is then sampled at the surface positions using trilinear interpolation. The current implementation simply projects the sampled values onto the x, y plane.

$$curved[x, y] = Interpolate(mpr\_data, Position = x, y, K_z(x, y))$$

This works well for a relatively smooth surface, but does not preserve true distances. A flattening technique would be superior especially for vessels that are very tortuous.

8.4.4 Options for the Curved Reformat

Options for the *Curved Reformat* module include the interactive setting of slice thickness and slice offset as well as the choice of the modality used while projecting the curved surface.

8.4.4.1 *Slice thickness and slice offset*

The slice thickness defines the width of the projection volume in direction of the elevation axis z' of the reformatted volume. The slice offset *n* (may be negative!) is added to the elevation coordinates K_z prior to the sampling.

8.4.4.2 *Projection type*

Types of projections include maximum (MIP), minimum (MinIP) and average intensity. Under the assumption that the vessel under inspection is fully and accurately defined with points along the center line, a MIP is not necessary and will even decrease the vessel-to-background distinction. However, an accurate and complete vessel definition is either very time-consuming (hand-picked centerline) or a very difficult problem. If the centerline is either incomplete or off-center, there is a risk of creating artificial stenosis, especially in case of the coronary arteries with small diameters (see Figure 17)

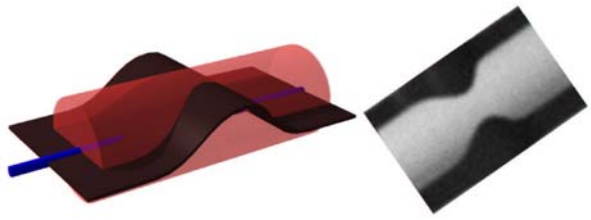


Figure 17: Stenosis introduced by an off-center vessel axis (from (Kanitsar, Fleischmann et al.))

Using a maximum intensity projection renders the reformatting process less sensitive to fluctuations in the vessel axis definition. Therefore, the curved reformat correspondents to

$$curved[x,y] = \max_{j=-s/2}^{s/2}\left(Interpolate(mpr\_data, Position = x, y, K_z(x,y) + j + n)\right)$$

where *s* is the slice thickness and *n* the slice offset. Thus no artificial stenosis is possible as long as the true central axis is within the range of the projection volume. Also, the MIP over a thin volume allows the visualization of previously undefined side branches and extensions.

8.5 Hessian-Matrix based Filtering

In order to facilitate vessel segmentation and vessel tracking, an enhancement filter is implemented. *Multiscale vessel enhancement filtering* based on all eigenvalues of the Hessian matrix was first proposed by (Frangi, Niessen et al. 1998) and was inspired by work from (Lorenz 1997; Sato, Nakajima et al. 1998). The excellent suppression of over-projecting organs as well as brilliant background noise reduction has since been confirmed in numerous papers (Frangi, Niessen et al. 1999; Wink, Frangi et al. 2002; Olabarriaga, Breeuwer et al. 2003; van Bemmel, Wink et al. 2003; van Bemmel, Viergever et al. 2004; Wink, Niessen et al. 2004)

The idea of the *vesselness filter* is to enhance voxels which have a high probability of being part of a vessel. To locally determine that likelihood, an eigenvalues analysis of the Hessian Matrix is performed. The Hessian matrix is defined here as

$$H(w) = \begin{vmatrix} L_{xx}(w) & L_{xy}(w) & L_{xz}(w) \\ L_{xy}(w) & L_{yy}(w) & L_{yz}(w) \\ L_{xz}(w) & L_{yz}(w) & L_{zz}(w) \end{vmatrix}$$

$L_{ij}(w)$ are the second-order partial derivatives of the Gaussian pre-filtered original data, where *w* denotes the width of the Gaussian kernel. Analyzing the eigenvalues of the Hessian matrix has an intuitive geometric justification in the case of vessel detection. For the further discussion, the eigenvalues λ_i shall be ordered in increasing magnitude ($|\lambda_1| \le |\lambda_2| \le |\lambda_3|$). If a bright vessel structure is present, then the eigenvalue belonging to the direction along the vessel is small, whereas the other two eigenvalues are large and negative. The corresponding eigenvectors u_i are an orthonormal triplet, with u_1 pointing in the direction of the vessel and $\{u_2, u_3\}$ forming a base to the cross-sectional plane orthogonal to the vessel direction.

Table 1 shows different possibilities for magnitude and sign of the eigenvalues, together with the pattern they represent.

Table 1: Possible values for the eigenvalues of the Hessian matrix (S = small, L+ = large and positive, L- = large and negative) and corresponding pattern.

λ_1	λ_2	λ_3	*pattern*
S	S	S	no preferred direction (noisy)
S	S	L -	bright plate-like structure
S	S	L +	dark plate-like structure
S	L -	L -	bright tubular structure
S	L +	L +	dark tubular structure
L -	L -	L -	bright blob-like structure
L+	L +	L +	dark blob-like structure

Based on the those observations, Frangi (Frangi, Niessen et al. 1998) introduces two geometric ratios (R_A, R_B) and a measure used to discriminate voxels belonging to the background (*S*). The first ratio is defined as $R_A = \frac{|\lambda_2|}{|\lambda_3|}$ and is able to distinguish between tubular-like ($R_A \approx 1$) and plate-like structures ($R_A << 1$).

The second ratio $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \cdot \lambda_3|}}$ is a measure for the deviation from a blob-like structure. R_B is maximal for a spherical structure, but cannot differentiate between a plane-like and a tubular-like structure.

The ratios defined above are grey-level invariant and therefore insensitive to intensity rescaling. However, we would like to include additional a-priori knowledge about the vessels, namely that they are brighter than the background and occupy a relatively small volume. If this information was not incorporated in the filter, background voxels would produce unpredictable filter responses due to random noise fluctuations. Therfore, we use the structure parameter $S = \sqrt{\sum_{i \le 3} \lambda_i^{\,2}}$ which is small in the background, where no structure is present. The three geometric parameters above are combined to a vessel enhancement filter together with the knowledge from Table 1 to

$$V(\vec{x}, w) = \begin{cases} 0, & if\ \lambda_2 > 0\ \ or\ \lambda_3 > 0 \\ v(\vec{x}, w), & otherwise \end{cases}$$

where $v(\vec{x}, w) = (1 - e^{-(R_A^{\,2}/2\alpha^2)}) \cdot (e^{-(R_B^{\,2}/2\beta^2)}) \cdot (1 - e^{-(S^2/2\gamma^2)})$

The parameters α, β, and γ adapt the filter response to the distinction between plate- and tubular-like structures (α, R_A), deviation from blob-like structures (β, R_B), and magnitude of background suppression (γ, S).

The filter response is a function of the Gaussian filter width *w*, therefore the filter is usually applied at a range of scales ranging from the smallest expected vessel width to the largest. In order to achieve a unique filter output for each voxel, the overall filter response is usually selected by

$$V(\vec{x}) = \max_{w_{\min} \le w \le w_{\max}} V(\vec{x}, w)$$

Filtering based on the eigenvalues of the Hessian matrix has been used as a preprocessing step for numerous applications, including maximum intensity projections, volume rendering, and automatic segmentation. In this work, the filtered data is only used as preparation for the automatic tracking of the vessel centerline. Even though the filter quality is excellent, the use of the filtered data for direct visualization is risky - mainly because the inclusion of a-priori knowledge about the vessel shape suppresses information that might have a high diagnostic value in pathological cases.

Because we use the filtered data for the vessel tracking, the filtering is applied at one single scale only, which corresponds to the expected vessel radius. In the current implementation, the best results have been produced with a Gaussian width *w=1*. The reduction to a single scale allows the efficient suppression of vessels with much larger diameters, for example the left or right ventricle. A 'leaking out' of the tracking algorithm is therefore prevented. The downside of this decision is that the filtered data misses the clinical relevance even more, since vessels with larger diameters are artificially diminished.

In order to assess the quality of the filter, the program allows the usage of the filtered data instead of the original MRA data for the multi-planar and curved reformats as well as for the segmentation step, the globe visualization, and the masked volume projection. Figure 18 shows a MIP and a curved reformat of the original data in comparison to the identical procedures based on the vessel-enhanced data.

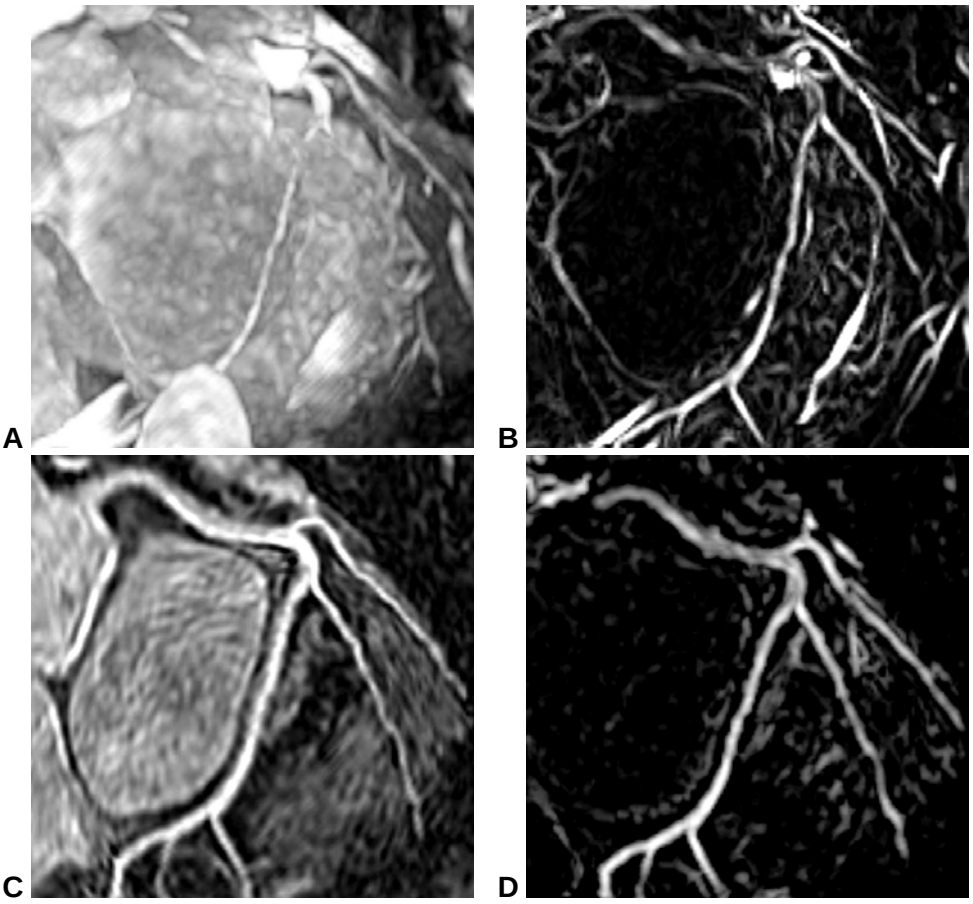


Figure 18: Vessel enhancement filter. In a MIP of the original data (thickness = 30 mm), the vessel is mostly disguised by signals from other organs (A). The MIP of the vessel-enhanced data shows the background suppression quality of the filter (B). However, other tubular structures are enhanced as well and occlude the view. (D) shows a curved reformat based on the filter-enhanced data. The morpholgy of the vessel is clearly visible, allowing an semi-automatic tracking. The curved reformat of the original data is shown in comparison (C).

8.5.1 Implementation

The eigenvalue computation of the Hessian matrix is a very expensive operation, especially for a large three dimensional dataset. Even though IDL is an interpreted language, operations are fairly efficient, at least as long as the input data is vectorized. However, an implementation strategy like the following is painfully slow:

```
FOR i=0, x_size-1 DO
  FOR j=0, y_size-1 DO
    FOR k=0, z_size-1 DO
       W= [[Lxx[i,j,k], Lxy[i,j,k], Lxz[i,j,k]]
           [Lxy[i,j,k], Lyy[i,j,k], Lyz[i,j,k]]
           [Lxz[i,j,k], Lyz[i,j,k], Lzz[i,j,k]]]
       eigen_values = LA_EIGENQL(W)
    ENDFOR
  ENDFOR
ENDFOR
```

The execution time amounts to 35 minutes for a 512 x 512 x 140 dataset, mainly because the triple FOR-loop has to be fully interpreted. Whereas the Gaussian filtering and the partial derivate computations are relatively simple (and could be done in the Fourier space as well), the bottleneck is the solving of the more than 36 million equation systems to find the eigenvalues. Even though the actual eigenvalue calculation is a highly optimized routine based on the LAPACK Linear Algebra Package (Anderson 1999), the overhead for calling this function takes roughly 95% of the execution time. There are three approaches to obliterate the need for the triple FOR –loop: *Tri-Diagonalization, External Function*, and *Algebraic Solution*, which are described in the following. The results for the filtering of a 512 x 512 x 140 dataset on a commercial desktop computer under Windows XP with a 3.0 GHz Pentium 4 Processor and 1GB of RAM (Dell Dimension 8300, Dell Computers, Austin, TX) are shown in Table 2. The current implementation uses the Algebraic solution, mainly because the minimal gain in using an external function is not worth the complexity this step involves.

The acquired filtering time of just under 4 minutes leaves clearly space for improvement, but might be sufficient for research purposes. The filtered data can be saved and restored, facilitating the repetitive processing of the same dataset. Software packages for clinical use are usually programmed in C, therefore obliterating the need for the approaches considered here.

Table 2: Comparison of different approaches to do the Hessian-based filtering in IDL

Approach	*Time needed for filtering*
Triple interpreted Loop (LA_EIGENQL)	35 minutes
Triple interpreted Loop (IDL EIGENQL)	68 minutes
Tri-Diagonalization (LA_TRIQL)	28 minutes
External Function (internal solver)	12 minutes
External Function (CLAPACK)	3 minutes, 25 seconds
Algebraic solution	3 minutes, 40 seconds

8.5.1.1 *Tri-Diagonalization*

The equation system of n different 3 x 3 matrixes can be transformed into a large tri-diagonal, 3n x 3n symmetric matrix. There are very efficient algorithms to find the eigenvalues of tri-diagonal large matrix. The LA_TRIQL procedure uses the QL and QR variants of the implicitly-shifted QR algorithm to compute the eigenvalues and eigenvectors of a symmetric tridiagonal array. However, the overhead of placing the sub-matrices and reassigning the eigenvalues is larger than the speed gained by this approach.

8.5.1.2 *External Function*

IDL allows the inclusion of external libraries, which can be written in C/C++, Fortran and many other languages. The compiled library functions can then be called from IDL, allowing an efficient processing of the dataset and providing a work-around for the interpreted loop limitation of IDL.

Whereas this approach promises the best results in theory, the actual implementation has been difficult. Depending on the internal library the C-Code uses for mathematical functions, the calculations are much slower than expected. In a first try, I used CLAPACK, a C-Implementation of the LAPACK package (http://www.netlib.org/clapack/). However, the LAPACK code is optimized for the solution of large eigenvalue systems, the overhead for calling the actual computation is large. After all, we are interested in efficiently finding the eigenvalues of a large number of 3 by 3 matrices, not the computation of a huge linear equation system. The second try was using a built-in eigenvalue solver, and the resulting computation time was even worse.

Key to the efficient solution would be a customized adaptation of a fast algorithm for small systems, e.g. based on Jacobi rotations (William H. Press 1988). The accuracy of the eigenvalue calculation may be chosen pretty low, especially since we are mainly interested in areas where the condition of the matrix should not be pathological.

8.5.1.3 *Algebraic solution*

Finding the eigenvalues of a 3 by 3 real matrix is congruent to the problem of root calculation of a third degree polynomial. Let H(w) be the Gaussian pre-filtered Hessian matrix.

$$H(w) = \begin{vmatrix} L_{xx} & L_{xy} & L_{xz} \\ L_{xy} & L_{yy} & L_{yz} \\ L_{xz} & L_{yz} & L_{zz} \end{vmatrix}(w)$$

where $L_{ij}(w)$ are the partial second derivatives. The eigenvalues λ are defined as

$$\left|H(w) - \lambda \circ I\right| = \begin{vmatrix} L_{xx} - \lambda & L_{xy} & L_{xz} \\ L_{xy} & L_{yy} - \lambda & L_{yz} \\ L_{xz} & L_{yz} & L_{zz} - \lambda \end{vmatrix} = 0$$

I is the identity matrix. The determinant can be reduced to

$$\lambda^3 + a \cdot \lambda^2 + b \cdot \lambda + c = 0$$

with

$$a = -L_{xx} - L_{yy} - L_{zz}$$

$$b = -L_{xy}{}^2 - L_{xz}{}^2 - L_{yz}{}^2 + L_{xx} \cdot L_{yy} + L_{xx} \cdot L_{xx} + L_{yy} \cdot L_{zz}$$

$$c = L_{xy}{}^2 \cdot L_{zz} + L_{xz}{}^2 \cdot L_{yy} + L_{yz}{}^2 \cdot L_{xx} L_{xx} \cdot L_{yy} + L_{xx} \cdot L_{xx} + L_{yy} \cdot L_{zz} - 2 \cdot L_{xy} \cdot L_{xz} \cdot L_{yz} - L_{xx} \cdot L_{yy} \cdot L_{zz}$$

We can solve this analytically (see e.g. (William H. Press 1988)). First compute $Q = (a^2 - 3 \cdot b)/9$ and $R = (2 \cdot a^3 - 9 \cdot a \cdot b + 27 \cdot c)/54$

If Q and R are real (always the case for a real symmetric matrix) and $R^2 \leq Q^3$ then the cubic equation has three real roots. We find them by computing

$$\phi = \arccos\left(R / \sqrt{Q^3}\right)$$

The three roots then are

$$\lambda_1 = -2 \cdot \sqrt{Q} \cdot \cos\left(\frac{\phi}{3}\right) - \frac{a}{3}$$

$$\lambda_2 = -2 \cdot \sqrt{Q} \cdot \cos\left(\frac{\phi + 2\pi}{3}\right) - \frac{a}{3}$$

$$\lambda_3 = -2 \cdot \sqrt{Q} \cdot \cos\left(\frac{\phi + 2\pi}{3}\right) - \frac{a}{3}$$

If $R^2 > Q^3$, then some of the roots have an imaginary part, usually one real root and one complex conjugate pair. Since the value of the roots have a physical meaning - the value of the eigenvalue indicates a change in curvature – we are only interested in the real solutions. We therefore set the eigenvalues to zero if $R^2 > Q^3$.

Analytically calculating the eigenvalues of the Hessian has the advantage that it can be fully vectorized, therefore allowing the use of IDL's very fast mathematical libraries. To limit the memory needed to do the computations, the actual filtering is done in smaller blocks. This is possible because the Hessian-based filtering depends only on the local structure. The current block size is 80 x 80 x 80.

8.5.2 Options for the Hessian-Matrix based Filtering

Options for the *Filtering* module include the possibility to limit the data used in the filtering process and to fully control the parameters employed for the vessel enhancement.

8.5.2.1 *Data range*

The filtering process can be restricted to a subset of the original data. Options include a manual delimitation or the restriction to a volume that encompasses the points on the vessel de-

fined so far. Also, the filtering can be done on a per-voxel base, which is useful if the needed filter-volume is much smaller than the overall data size.

8.5.2.2 *Filtering type*

The parameters α, β, and γ are either set manually, or default values for an optimal tracking can be used. Best results for tracking purposes are achieved with $\alpha = 0.10$, $\beta = 0.72$, and $\gamma = 0.36 \cdot I_{\max}$, where $I_{\max}$ is the maximum intensity value of the original data.

Values found in other publication differ from these values, e.g. $\alpha = 0.25$, $\beta = 0.25$, and $\gamma = 0.38 \cdot MAX(data)$ in (van Bemmel, Viergever et al. 2004). Our settings weight the importance of the distinction between plate- and tubular-like structures (α, R_A) less than the deviation from blob-like structures (β, R_B) in order to facilitate an accurate vessel tracking. The object of the paper referenced above was to do a tracking and segmentation of the *internal carotid artery*, and was aimed at automatic stenosis quantification with no branching involved. For the coronary tree, branching of the vessel is frequent and should not result in signal loss at junctions. The local curvature structure at a branching point is closer to a plane than to line, explaining why we achieved better results with a lower α value. On the other hand, testing has not been extensive, so these results should be interpreted cautiously. Figure 19 shows a curved reformat of the original data filtered with different α, β, and γ values.

Figure 19: Curved reformat of the filtered data. (A) shows the original image. (B) shows the curved reformat of data filtered with $\alpha = 0.25$, $\beta = 0.25$, and $\gamma = 0$. There is considerable noise due to fluctuations in the backround structure which are no longer suppressed by the filter. (C) shows the filter output with the optimal parameters for the purpose of tracking ($\alpha = 0.10$, $\beta = 0.72$, and $\gamma = 0.36 \cdot I_{\max}$). The data range for the filter has been restricted to a volume enclosing the defined vessel center line.

8.6 Vessel Tracking

The aim of the vessel tracking module is to relieve the human operator of the tedious alignment and manual entering of points belonging to the coronary tree. The problems of vessel tracking and vessel segmentation are closely related. In general, it is possible to achieve accurate vessel segmentation with knowledge of the vessel axis, e.g. based on a local region growing algorithm (Yi and Ra 2003) or with a model-based approach (Frangi, Niessen et al. 1999). The opposite way of obtaining the vessel tree from segmented data is known as three-dimensional thinning and is usually based on morphological hit-or-miss operations.

However, if neither segmentation nor vessel axis is known a-priori, the robust and accurate tracking of a vessel tree can be very demanding. An overview of methods with emphasis on vessel tracking and segmentation of the human leg can be found in (Felkel and Wegenkittl). *Direct vessel tracking*, an approach based on a *minimum cost path*, and *wavefront propagation* have been examined with respect to coronary MRA-data within the framework of the CoroViz software.

8.6.1 Direct Vessel Tracking

Direct vessel tracking as proposed by (Wink, Niessen et al. 2000) describes an interactive and local vessel tracking method. The operator first defines two starting points in the thick part of the vessel as an indication of the initial vessel direction. The method then estimates the position of the next candidate point in this direction based on the current minimal vessel diameter. A fan of rays is casted from each point in a square perpendicular to the current vessel direction. The rays end at the vessel border and therefore allow in theory the precise positioning of the new point as the center of mass from all rays.

Even though Wink et al. introduced variants to force a search direction, limit the curvature, and even to detect branching, the algorithm does not work very well in our case. Direct vessel tracking has been tested mainly on CTA and MRA data of the abdominal aorta, where the vessel cross section is far larger than in our case. Our testing on coronary MRA data suggests that the algorithm gets off track either as soon as the cross section area is less than about 16 pixels or if there are bright structures from other organs close-by.

8.6.2 Minimum Cost Path

The *minimum cost path* (MCP) approach, as originally proposed by (Dijkstra 1959), interconnects two or more user-defined points on a path with minimal cost, defined as the sum of all transition costs involved in the traversal. The transition costs of traveling from one node to its neighbor is given by a cost function which may be dependent on the data, its gradient, or any other measure. Different cost functions result in different MCPs.

A cost function based on the reciprocal output of the Hessian-based vessel enhancement filter has successfully been used for MRA coronary axis determination (Wink, Frangi et al. 2002) and for the semiautomatic segmentation of the internal carotid artery (van Bemmel, Viergever et al. 2004), also with MRA data. A search tree is initiated from the start node, and the propagation is continued until it has reached all other defined points. The MCP is then found by backtracking. Alternatively, the search process may be started at all points simultaneously and is continued until the points have been connected. The later approach is more efficient, but requires a careful handling of the search process to guarantee that the global minimum path is eventually found. Further

elaborations are found in (Olabarriaga, Breeuwer et al. 2003), and a comparison of the MCP-tracked centerline with manual tracings was shown in (van Bemmel, Wink et al. 2003).

The main drawback of the MCP approach is that it only connects end points on a vessel structure. In the case of coronary MRA, the distal part of the vessel structure is not known a-priori. Finding all the endpoints of the tree-like structure would be a very lengthy process – the human operator would have to track down the vessel starting from the root, which is exactly the part we would like to automate.

8.6.3 Wavefront Propagation

A method of region growing with simultaneous graph generation was presented by (Zahlten 1995). The underlying idea is very similar to the MCP generation described above. Starting from an initial seed point, a wave is initialized and travels through the objects. Neighboring voxels are included in the wavefront as long as they fulfill local and/or global requirements. The neighboring relationship can be defined as a 26-node, 18-node, or 6-node neighborhood. Depending on the chosen neighborhood, the search process embraces and analyzes the 26, 18, or 6 closest voxels. Figure 20 shows the differing neighborhood definitions.

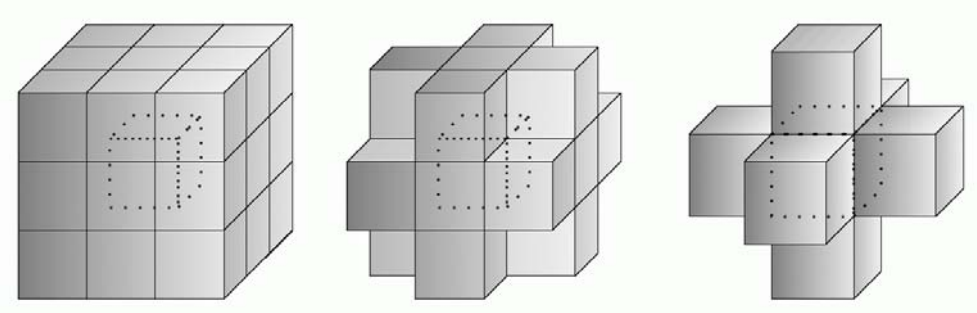


Figure 20: Different neighborhood definitions for the central voxel. (A) 26-node neighborhood, (B) 18-node neighborhood, (C) 6-node neighborhood

The main difference between a classical region growing and the method proposed by Zahlten lies in the inclusion of a bifurcation detection and the generation of the symbolic vessel tree in parallel to the wave propagation. As soon as the newly included voxels are no longer mutually connected, a branching point is detected and the symbolic tree is updated with the bifurcation/n-furcation information. Each new branch is labeled, with the longest one holding the current label. The wave is then updated in order of branch length, resulting in a top-to-bottom tree generation.

Zahlten also proposes a wave correction algorithm to assure that the wave propagation is always perpendicular to the current vessel direction. This help to avoid artificial branching detection in places with high vessel curvature. Different filtering methods are used as preprocessing step, namely median and locally adaptive filtering to preserve thin connections.

8.6.4 CoroViz Tracking

The current version of the CoroViz tracking algorithm implements an adapted version of Zahlten's wavefront and region-growing approach as described above. The current implementation uses the vessel enhancement filter as a preprocessing step. Wave correction and the generation of a symbolical tree are not yet supported.

Main advantages are:

- Inherent branch detection capability
- Wave propagation process is closely related to the actual physical blood flow inside the vessel ('where the blood flows, the wave will pass as well')
- Fast implementation possible compared to expensive 3D thinning algorithm
- The operator can easily add new seed points or manually correct the found centerline.

Figure 21 shows the different steps in the tracking procedure. Inputs of the tracking algorithm are the filtered data and at least one user defined point on the coronary vessel. Figure 21-A shows a curved reformat of the filtered data.

The global inclusion criterion for the wave propagation is a seed-point dependent threshold, corresponding to the following wave propagation speed function

$$c(\vec{x}) = \begin{cases} 1, & if \quad filtered\_data(\vec{x}) < THRESHOLD\_\% \cdot filtered\_data(\vec{s}) \\ 0, & else \end{cases}$$

where *filtered_data* is the vessel-enhanced original data, *THRESHOLD_%* a user-defined relative threshold value, and $\vec{s}$ the seed point coordinates.

Figure 21-B shows the thresholded filtered data with the seed point highlighted. Based on the seed point, a wave is initialized. The wave front then propagates according to the speed function $c(\vec{x})$. The center point of a wave front is defined as either the center of mass or the maximum data value of all points belonging to the front. As soon as the wave gets to a branching of the vessel (see Figure 21-C), the wave front is no longer connected and therefore the center points are calculated for each vessel branch separately. All the points belonging to the wave so far are marked brighter then the thresholded volume.

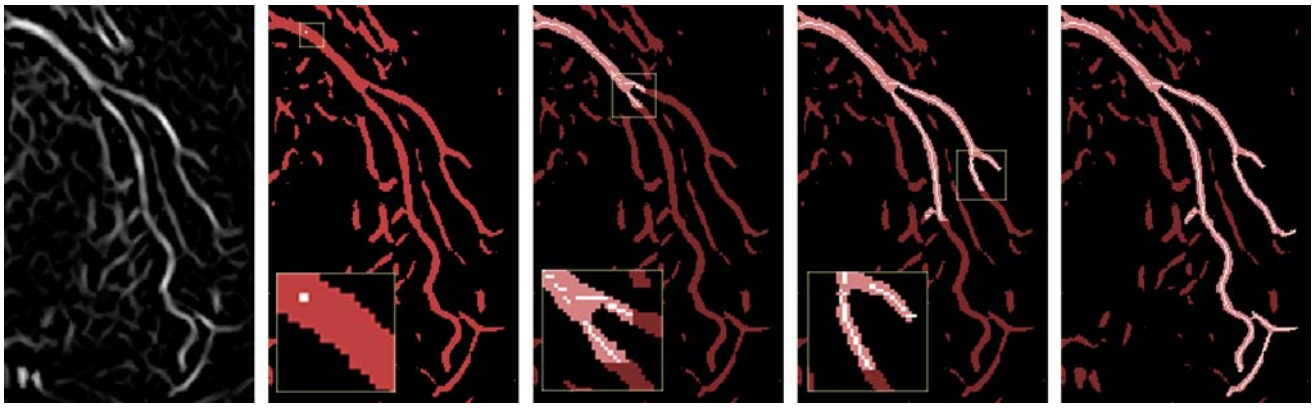


Figure 21: CoroViz tracking algorithm. (A) shows a two-dimensional curved reformat of the filterered data. The inclusion criteria is chosen as a relative threshold (30 % of the seed point value), with the seed point shown as a bright spot (B). The wave propagates starting from the seedpoint. The tracked centerline is marked with white dots (C), the hitherto visited area is highlited. Both dead ends and branching of the vessel are detected (D). Note that the centerline is not connected. (E) shows the tracked centerline prior to tree pruning.

8.6.4.1 *Implementation*

The tracking is carried out separately for the RCA, the LAD, and the LCX. Listing 1 shows the pseudo-code. The complete pseudo-code listings can be found in the Appendix.

Listing 1: Main algorithm used for the CoroViz Vessel tracking

```
ALGORITHM TrackCenterTree
[t01]   FOR (all seed points) DO
[t02]      initialize wavefront with seed point
[t03]      WHILE (wavefront NEITHER empty NOR too large) DO
[t04]          UpdateWaveFront
[t05]          SplitWaveFrontIntoRegions
[t06]          FOR (all regions in regionslist) DO
[t07]                FindCenterOfRegion
[t08]               add center to center_line
[t09]          ENDFOR
[t10]      ENDWHILE
[t11]      move all points from center_line to center_tree
[t12]   ENDFOR
[t13]   PruneCenterTree
ENDALGORITHM
```

The region-growing process is started from each seed point [t01]. The wavefront propagates until no additional voxels can be found. Also, if the wavefront explodes (e.g. due to a low threshold), the propagation is stopped [t03] to prevent exhaustive computations. The current

wavefront is split into different regions [t05], and the center voxel is found for each region separately [t07]. The new center voxels are then added to the previously found center line [t08].

Since the region-growing process is started from each point separately, the resulting coronary tree will have duplicate branches, especially if the seed points are close together and the search regions as defined by the maximum search distance overlap. A pruning of the tree is therefore necessary [t13].

It is important to note that the current implementation treats the center line as well as the center tree simply as a set of points. The points are not structured, so there is no explicit information on the position of branching or end points available. Simply put, there is no symbolic tree generated during the tracking process.

The advantage of this approach is that the found vessel points are similar to the ones defined by the operator manually. The tracking module therefore fits nicely into the current workflow. Refinement and alignment of the points is possible without the need to update the symbolic tree. However, the unavailability of morphologic-geometrical information also restricts the further use for advanced visualization (see e.g. the masked volume projection module).

8.6.4.2 *Options for CoroViz Tracking*

Options for the tracking algorithm include the choice of the percentage value used for thresholding the data, the maximal search distance, and the selection of data source and seed points used in the tracking algorithm.

The *threshold value* determines how bright a voxel has to be (in relation to the value at the seed point) in order to be considered as part of the vessel structure. The *maximal search distance* is used to limit the search area to a cubic volume around the used seed points to a sub volume of the whole dataset. This option is used mainly to limit the maximal possible search tree and therefore prevent the computation of unneeded branches.

For testing purposes, the tracking may be done based on the original data as well the filtered data. Also, if there is no filtered data currently present, the vessel enhancement filtering can be done on the fly for the investigated voxels only. Even though the tracking algorithm is slowed down, this approach is still more efficient than a previous filtering of the whole dataset.

So far, we assumed that the tracking is done with only a few user defined points. However, an iterative tracking is possible by using points on the found center line as the new seed points for the next wave propagation. Since the threshold value depends on the value at the given seed point position, the search process adapts to the local brightness distribution.

It is possible to use all points defined, only the detected end points, or the search process can be limited to the last defined single point. The use of all points - together with a small maximum search distance to limit computational expense – is very useful in detecting additional side branches. A limitation to the end points works well for the distal elongation of vessel branches. An end point is detected either if there is no other point close by, or if the difference between the two closest points is larger than the distance in vessel direction between tracking points. The last-point-only mode is used frequently if the operator defines one additional point on the coronary vessel and would like to start a new tracking from there.

8.7 Vessel Segmentation and Surface Creation

Vessel segmentation is the process of distinguishing between voxels belonging to the vessel and voxels belonging to the background. Based on this knowledge, the vessel surface can be computed as a three-dimensional contour that separates vessel voxels from background voxels. Figure 22 shows an example of a three-dimensional scene with the segmented coronal tree depicted from different directions. Different automated vessel segmentation methods are addressed in (Falcao and Udupa 2000; Lakare 2000; Bülow, Lorenz et al. 2003; de Koning, Schaap et al. 2003; Yi and Ra 2003).

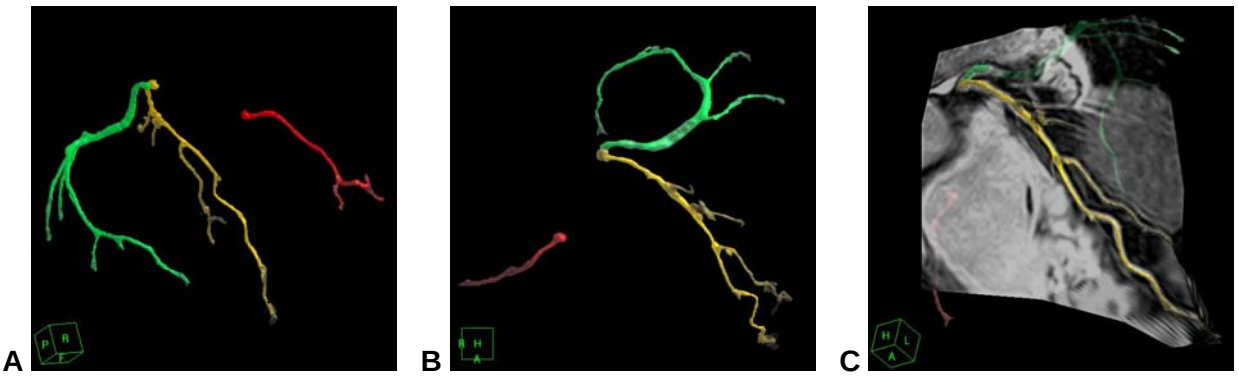


Figure 22: Vessel segmentation and surface creation. (A) and (B) show the segmented coronal tree from two different view angles. A transparent three-dimensional curved reformat is added, allowing a comparison of the vessel lumen between the two visualization modules.

8.7.1 Segmentation

Segmentation of the coronary arteries is difficult, especially in the case of MRA data where other bright structures are close by and the vessels are thin and tortuous. While results look appealing from a graphical point of view, clinical use is limited. The vessel segmentation module in the CoroViz software is mainly used in comparison with the volume projection and other visualization modules and does not claim any clinical significance besides an indication on the general vessel position.

The implemented segmentation algorithm relies on the knowledge of the vessel centerline as defined by the tracking algorithm. A vessel segmentation of the internal carotid artery based on the vessel axis and a level-set technique is described in (van Bemmel, Viergever et al. 2004). In (de Koning, Schaap et al. 2003), a vessel model with a *non-uniform rational B-spline* (NURBS) surface is deformed and matched to the underlying data. Whereas the achieved results look very realistic, the inclusion of a-priori knowledge again jeopardizes clinical use in pathological cases.

The CoroViz segmentation is a simple vessel lumen segmentation based on a three-dimensional variation of the *full with at 30 % maximum criterion* (FW30%M) as used in (Westenberg, van der Geest et al. 2000). Basically, all voxels surrounding the center point that have at least an intensity of 30 % are considered to be part of the vessel. However, Westenberg et al. used the criterion for the measurements of peripheral arteries based on gadolinium contrast-

enhanced MRA data. The coronary MRA data are acquired without the administration of an contrast agent, so the percentage value has to be adapted to a higher value. Also, to prevent a leaking out of the segmentation, the distance of an included voxel is limited to the maximum expected vessel radius. Figure 23 shows the CoroViz segmentation of an RCA with different parameters. Best results are achieved with a threshold value of 80 %(corresponding to a full width at 80% maximum criterion) and a maximum vessel radius of 2.5 mm.

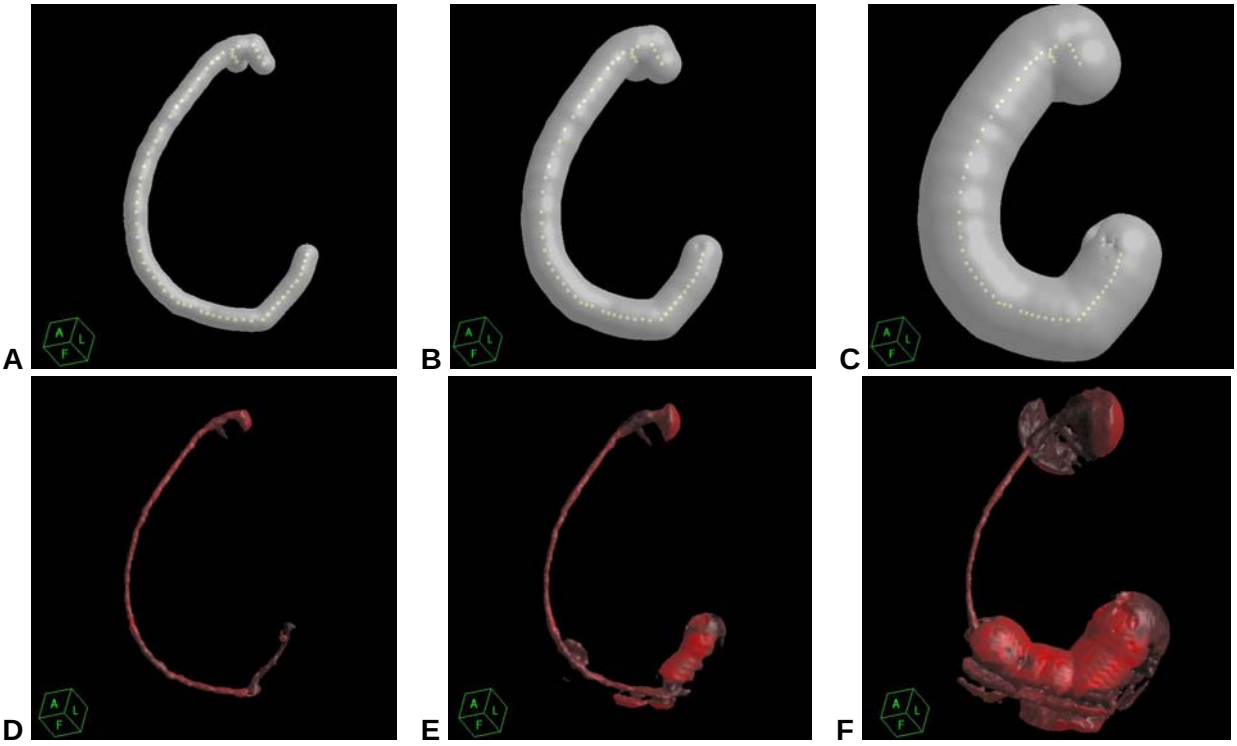


Figure 23: CoroViz Segmentation with limitation of maximal vessel radius. (A,B,C) show the search region for a maximal vessel radius of 2.5 mm, 5 mm, and 10 mm. (D,E,F) show the resulting segmentation for a threshold of 80 % center value. In (E, F) the leaking out of the region-growing segmentation is clearly visible.

8.7.2 Surface Creation

For the creation of contour surface between vessel voxels and background voxels a surface construction algorithm called *marching cubes* can be employed (William E. Lorensen 1987). It uses a divide-and-conquer approach and is able to maintain inter-slice connectivity, surface data, and gradient information present in the original 3D voxel based data. The algorithm first determines how the surface intersects this cube, then *marches* to the next cube.

In summary (William E. Lorensen 1987), the surface from a three-dimensional set of data is created in the following way:

- Create a cube from four neighbors on two adjacent slices
- Calculate an index for the cube by comparing the eight density values at the cube vertices with the surface constant.

- Using the index, look up the list of edges from a precalculated table. Figure 24 shows the 15 different possibilities.
- Using the densities at each edge vertex, find the surface-edge intersection via linear interpolation.

The triangle vertices and indices are returned and can be visualized as a surface mesh with or without shading.

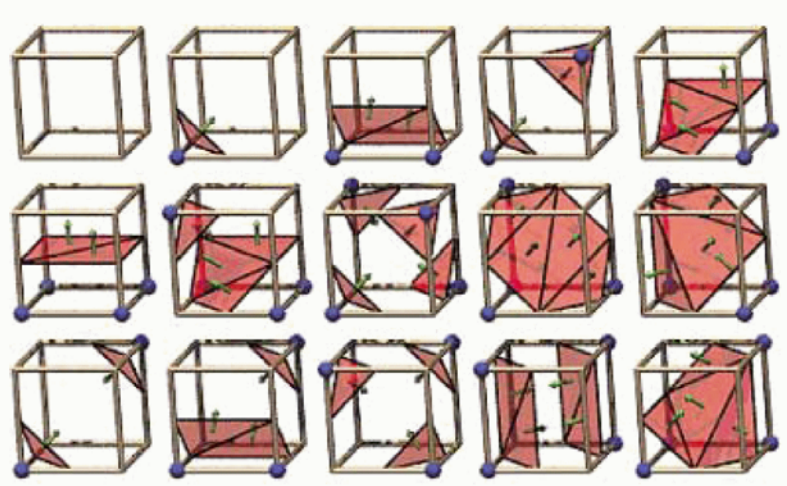


Figure 24: The 15 possible intersection of an iso-surface depending on the 8 neighboring voxels (William E. Lorensen 1987).

8.7.3 Implementation

For the implementation of the local region-growing segmentation, the SEARCH3D procedure is employed to find voxels belonging to the vessel. Given a starting location and a range of values to search for, SEARCH3D finds all the cells in the search volume that are within the specified range of values and have some path of connectivity through these cells to the starting location.

The SEARCH3D procedure is called for every seed point with a limited search region indicated by the maximum search distance and the lower threshold set to a given percentage of the center voxel. The position of all the voxels that are part of the vessel is then marked in a three dimensional volume for the processing with the SHADE_VOLUME procedure.

SHADE_VOLUME produces a list of vertices and polygons describing the contour surface. It computes the polygons that describe the contour surface by visiting each voxel to find the polygons formed by the intersections of the contour surface and the voxel edges. This method is similar to the marching cubes algorithm described above (William E. Lorensen 1987). The vertices and polygons can then be rendered and are part of the CoroViz three-dimensional scene.

8.7.4 Options for the Vessel Segmentation

Options for the vessel segmentation include the maximum search distance and the relative threshold as described above as well as the possibility to smooth the mesh grid that depicts the

vessel wall. The smoothing function performs spatial smoothing on the polygon mesh output of the Marching Cubes algorithm by applying Laplacian smoothing to each vertex, as described by the following formula:

$$\vec{x}_{i,n+1} = \vec{x}_{i,n} + \frac{\lambda}{m}\sum_{j=0}^{m}\left(\vec{x}_{j,n} - \vec{x}_{i,n}\right)$$

where $\vec{x}_{i,n}$ is the vertex *i* for iteration *n*, λ is the smoothing factor, and *m* is the number of vertices that share a common edge with $\vec{x}_{i,n}$,

It is important to realize that a smoothing of the mesh grid will usually result in a slimmer appearance of the vessel structure. Figure 25 shows a section of the LCX with different settings for the smoothing of the mesh grid.

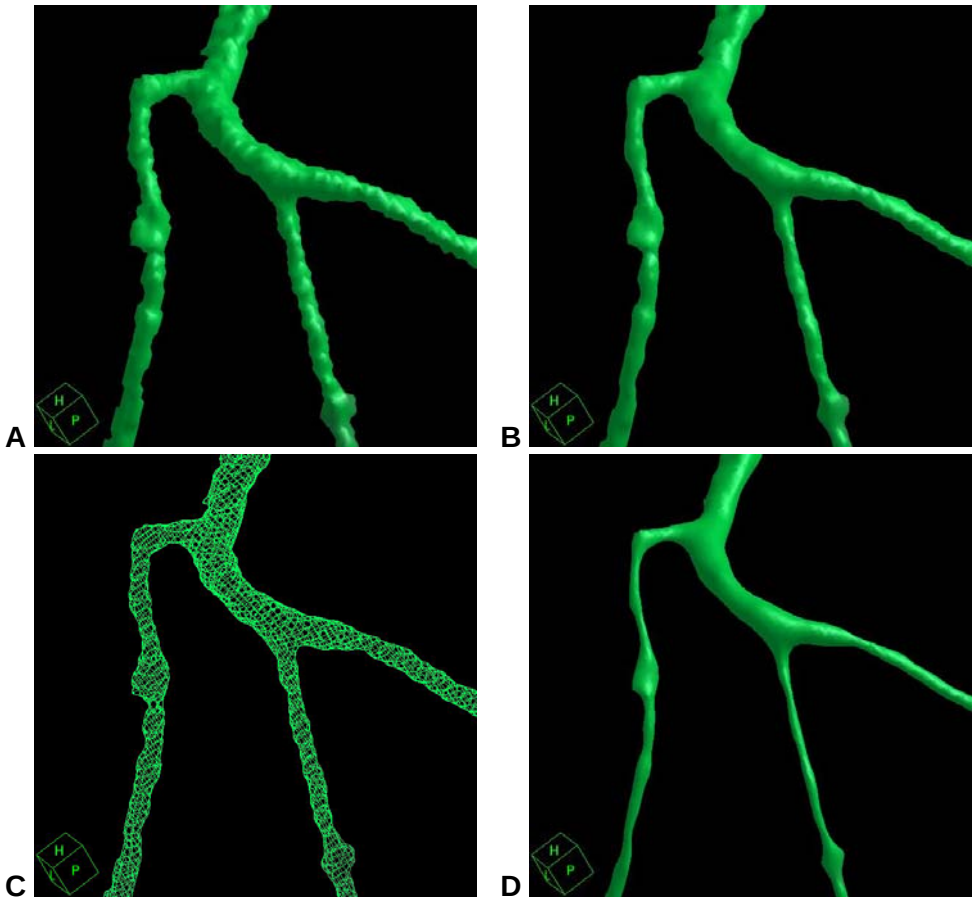


Figure 25: CoroViz segmentation (threshold = 80% center value, maximum search distance = 5) of a branching distal part of the LCX. (A) shows the shaded output of the Marching Cubes algorithm. The smoothed output after 40 iterations with smoothing factor λ = 0.04 is shown as a rendered shaded surface (B) and as wire exposition (C). After 100 iterations with the same smoothing factor, the branches are thinned out and even an unwanted break-up is possible (D).

8.8 Globe Visualization ('GlobeViz')

The aim of the globe visualization is to deform a sphere so that all points belonging to the coronal tree lie on its surface. This concept is closely related to the original Soap-Bubble idea. (Etienne 2001; Etienne, Botnar et al. 2002). However, instead of flattening the surface by means of a parallel projection, we'd like to keep the three-dimensional model, allowing the operator to rotate, zoom, and pan the image data so the chosen perspective is optimal for the vessel under investigation.

Surface creation and globe projection are very similar to the concepts employed for the curved reformat. However, instead of bending a plane, we deform a sphere, and instead of projecting in z-direction, we calculate the MIP in radial direction. Figure 26 shows the sphere as a mesh grid before and after the deformation.

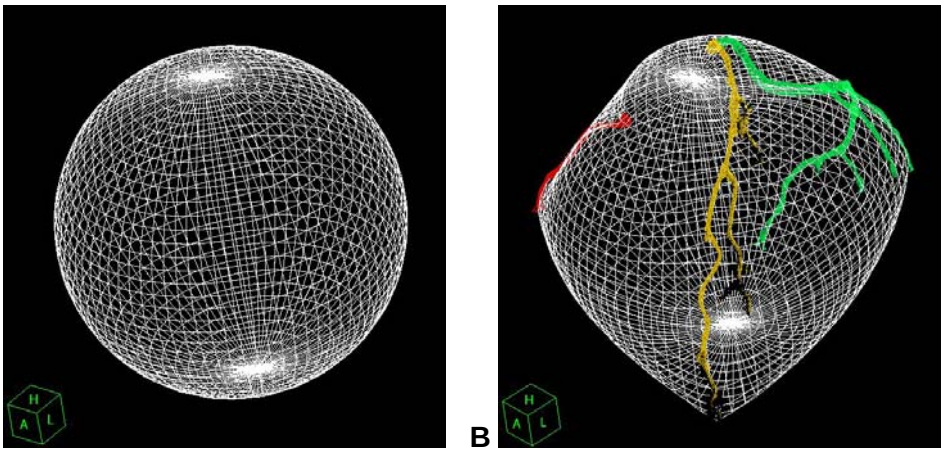


Figure 26: Globe Vizualization. A sphere is locally deformed in a way that the coronary anatomy lies on its surfcace.

A sphere is fully defined by the specification of a center $P_0 = \{X_0, Y_0, Z_0\}$ and an initial radius r_0. Let $P_{i,RCA} = \{X_{i,RCA}, Y_{i,RCA}, Z_{i,RCA}\}$ be the set of n_{RCA} points defining the right coronary artery, $P_{i,LAD} = \{X_{i,LAD}, Y_{i,LAD}, Z_{i,LAD}\}$ the set of n_{LAD} points defining the LAD, and $P_{i,LCX} = \{X_{i,LCX}, Y_{i,LCX}, Z_{i,LCX}\}$ the set of n_{LCX} defining the left circumflex artery.

As the center of the sphere we choose

$$P_0 = \frac{1}{3}\left(\frac{1}{n_{RCA}} \sum_{i=1}^{n_{RCA}} P_{i,RCA} + \frac{1}{n_{LAD}} \sum_{i=1}^{n_{LAD}} P_{i,LAD} + \frac{1}{n_{LCX}} \sum_{i=1}^{n_{LCX}} P_{i,LCX} \right)$$

The equal weighting of the three branches assures that the center of the sphere lies in between the vessels even if the number of defined points is different for each vessel.

Let $P_i = \{P_{i,RCA}, P_{i,LAD}, P_{i,LCX}\}$ the set of all points, and $n = n_{RCA} + n_{LAD} + n_{LCX}$ the total number of points defined. We select an initial vessel radius of

$$r_0 = \frac{1}{n}\sqrt{(X_i - X_0)^2 + (Y_i - Y_0)^2 + (Z_i - Z_0)^2}$$

A coordinate transform is needed to convert all Cartesian coordinates to spherical coordinates. The points $P_i = \{X_i, Y_i, Z_i\}$ are transformed to $P_i' = \{lon_i, lat_i, r_i\}$, where lon_i is the longitude, lat_i is the latitude and r_i is the radius at the given point.

The relationship to geodesy is apparent. We can see the deformed sphere as a deformed globe. Therefore, we use methods from geodesy to first fit a surface through the irregularly defined points and then flatten it.

In geodesy, longitude and latitude are the coordinates used in designating the location of places on the surface of the earth. Latitude, which gives the location of a place north or south of the equator, is expressed by angular measurements ranging from 0° at the equator to 90° at the poles. Longitude, the location of a place east or west of a north-south line called the prime meridian, is measured in angles ranging from 0° at the prime meridian to 180° at the International Date Line. Degrees of latitude are generally equally spaced, yet vary to a slight degree due to the flattening at the poles. The meridians with constant longitudes are farthest apart at the equator and converge towards the poles.

The points $P_i' = \{lon_i, lat_i, r_i\}$ can be compared to local measurements of longitude lon_i, latitude lat_i, and elevation r_i. The object is to find a surface that is both smooth and intersects the points $\{lon_i, lat_i\}$ at elevation r_i. This process is generally called triangulation.

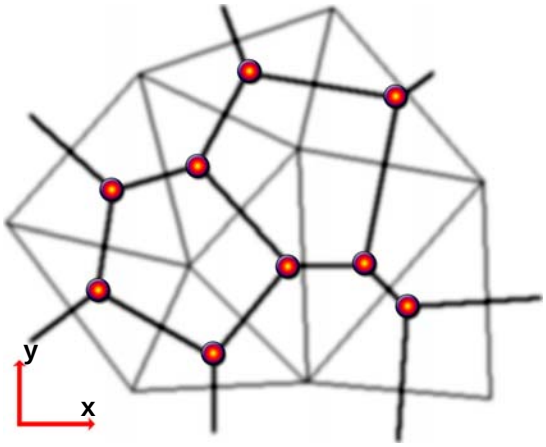


Figure 27: Delauney triangles (thin lines) and associated Voronoi tesselation (thick lines) for nine input points.

Triangulation involves creating a set of non-overlapping triangularly bounded facets from the sample points, with the vertices of the triangles being the input sample points. The Delauney triangulation is closely related to the Voronoi tessellation, which splits the plane into a number of polygonal regions ('tiles'). Each region has one sample point in its interior. All other points inside the polygonal tile are closer to the generating point than to any other. The Delauney triangulation is created by connecting all generating points which share a common tile edge. Figure 27 shows a Delauney triangulation together with the associated Voronoi tessellation.

Once all the triangles are defined, the elevation coordinates r_i may be interpolated for the desired grid coordinates $\{longrid, latgrid\}$ using e.g. *linear*, *quintic*, *minimum curvature*, *kriging* or *thin-plate spline interpolation*. The values of $r\ (longrid, latgrid)$ depicts the elevation as a function of the longitude and latitude coordinates of the grid array.

8.8.1 Globe Viz Projection

The interpolated points $\{longrid, latgrid, r(longrid, latgrid)\}$ define a surface through the original data volume. First, the interpolated points are transformed back into the Cartesian coordinate system. The data is then sampled at the surface positions using trilinear interpolation.

$$surface[longrid, latgrid] = Interpolate(data, Position = longrid, latgrid, r(longrid, latgrid))$$

surface is two-dimensional array, corresponding to a sampled gray-value image depending on longitude and latitude. In continuing analogy to geodesy, *surface* would be a map of the coronaries. We can then wrap the map around the three-dimensional globe model, a process called *texture mapping*. Figure 28 shows the analogy between covering the earth globe with a map of the continents and warping the surface image to the coronary globe.

It is important to note that the depicted map projections do not preserve area or length. Due to the fact that the meridians converge, the spatial sampling is much denser towards the poles. Therefore, the areas with latitude close to +/- 90 degrees appear stretched on the two-dimensional maps. However, since the texture image is projected back to the original sampling surface, length and area are again preserved in the three-dimensional model.

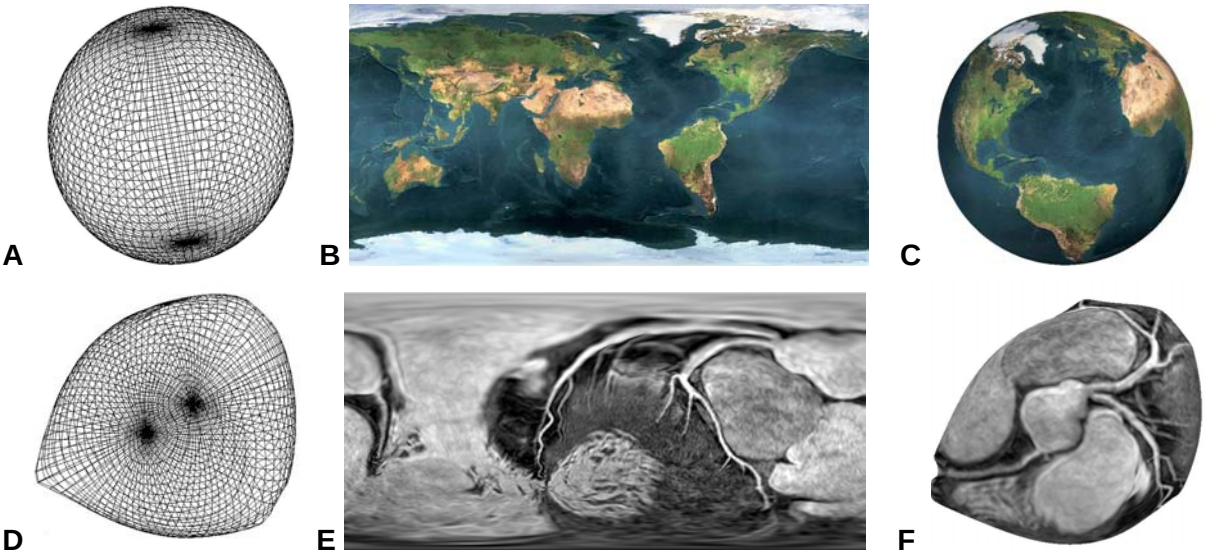


Figure 28: Analogy between GlobeViz texture mapping and earth map projection. The earth map (B) is projected onto a sphere mesh (A), resulting in a three-dimensional model of the earth globe (C). Similarly, a map of the coronaries (E) can be warped onto the interpolated surface grid (E). The resulting GlobeViz model is also three-dimensional and preserves length and area.

8.8.2 Implementation

The GRIDDATA function can be used to interpolate scattered data values on a sphere to a regular grid. This is accomplished using one of several available methods. Interpolation methods supporting interpolation on a sphere include:

- Inverse Distance
- Natural Neighbor
- Kriging
- Nearest Neighbor
- Radial Basis Function

Best results have been achieved with the Kriging method.

8.8.3 Options for the Globe Visualization

Options for the Globe Visualization module include the definition of grid and texture image resolution, as well as a choice of the model used to project the surface image on. The globe surface can be inflated or deflated with or without a restriction to regions outside the vessels. Finally, it may be useful to limit the set of used points to a sub-set of all vessels used for globe visualization procedure.

8.8.3.1 *Grid Resolution*

The grid resolution defines the number of circles (longitude and latitude) used in creating the surface. If the grid resolution is too low, important vessel features might not be depicted because the resulting sampling surface does not pass through all defined vessel points. Figure 29 shows the influence of grid resolution on image quality.

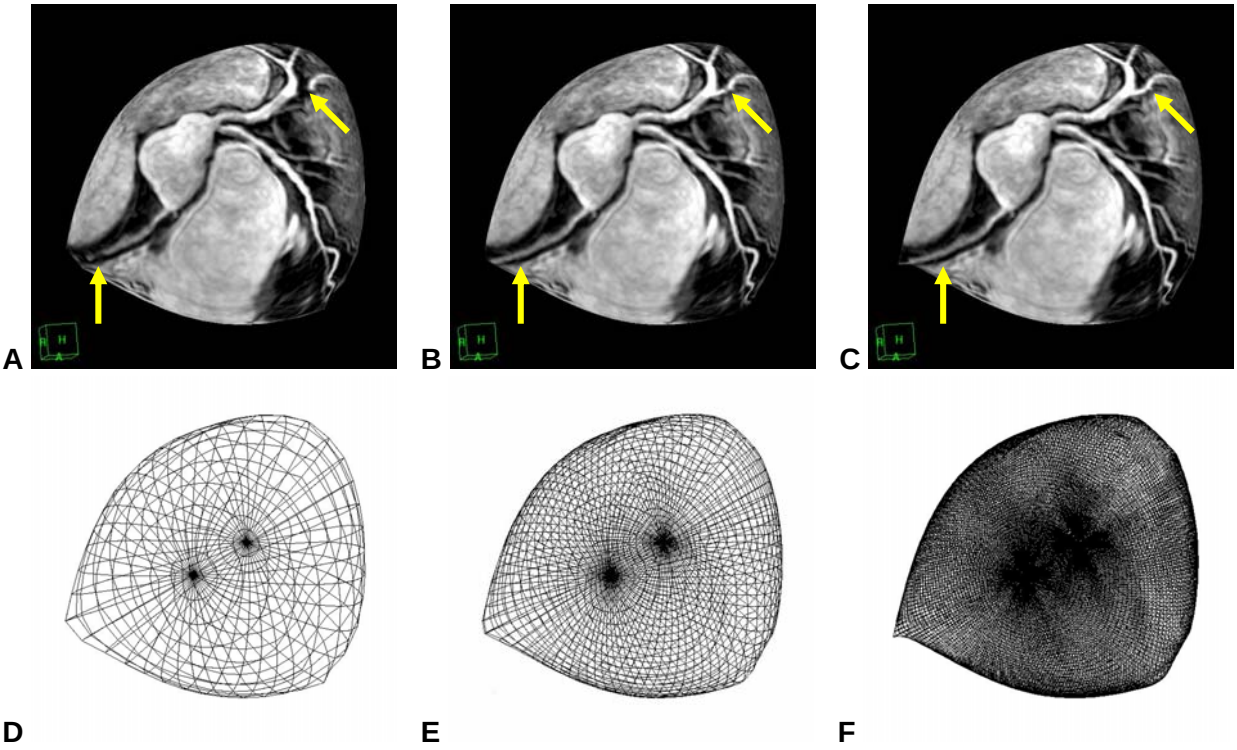


Figure 29: The influence of grid resolution on image quality. The top row shows the resulting globe surface for a (longitude, latidude) resolution of 9 degrees (A), 4.5 degrees (B), and 1.5 degrees (C). (D,E,F) show the corresponding surface grids. In the case of a 9 degree resolution (A,D), image artefacts and loss of vessel depiction is clearly visible (see arrows).

8.8.3.2 *Texture Resolution*

The creation of the smooth surface grid is an expensive operation, especially if non-linear interpolation methods such as *kriging* or *thin-plate spline interpolation* are used. If the texture resolution is identical to the longitude/latitude grid used in the surface creation, the data is sampled at each grid point. However, the three-dimensional grid positions can also be linearly interpolated in order to sample at points in between the grid defined by the surface, resulting in a higher texture image resolution. The combination of low surface grid and high texture resolution is much

more efficient and yields better results than a medium surface grid resolution with no oversampling.

One has to distinguish between an interpolation on the resulting texture image and an interpolation on the sample points used in creating the texture image. The first does not change the information content, but simply makes the texture look smoother. The lather uses additional knowledge from the original three-dimensional data. To emphasize the differences for different texture resolutions, the texture images themselves are not interpolated in this section. Figure 30 shows the globe visualization for different texture resolutions.

Figure 30: The influence of texture resolution on image quality. The [longitute, latitude] texture resolution of (A) correspondents to [256, 128], (B) is [512,256], and (C)is [1024,512]. In (A) the pixel size of the texture image is clearly visible.

8.8.3.3 *Surface Type*

Instead of mapping the interpolated texture image on the deformed globe, it's also possible to project the image on the initial sphere with center $P_0 = \{X_0, Y_0, Z_0\}$ and radius r_0. Figure 31 shows a globe visualization together with the projection on the original sphere. It's important to note that true distances in the three-dimensional model are no longer preserved.

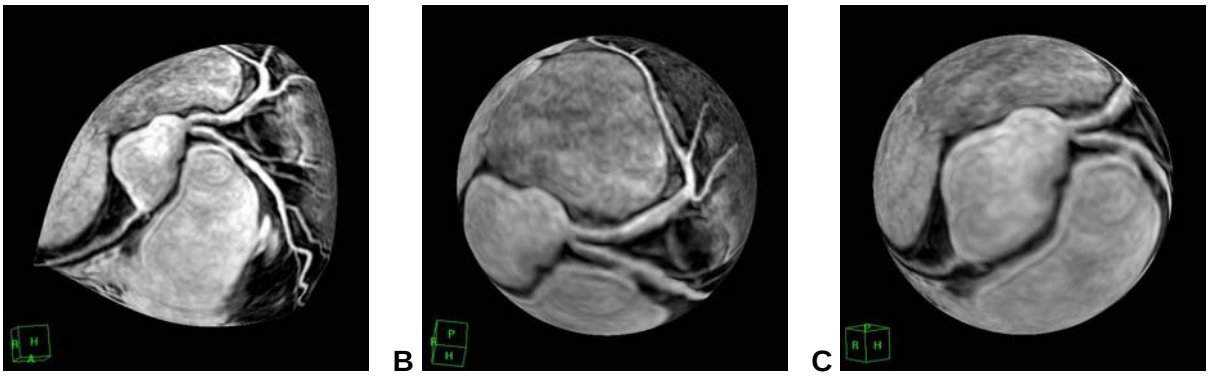


Figure 31: Sphere visualization. Instead of mapping the aquired image to the deformed surface (A), the texture is projected on a sphere with the original radius and center. (B,C) show the sphere MIP from two different view angles.

8.8.3.4 *Radial Offset*

The option of a radial offset allows the operator to grow and shrink the surface of the globe in radial direction (i.e. from and towards the center point). This forces an adaptation of the formula used when interpolating the texture image.

$$surface[longrid, latgrid] = Interpolate(data, Position = longrid, latgrid, r(longrid, latgrid) + offset)$$

where *offset* may be negative and indicates the desired offset from the original deformed globe. Figure 33 depicts the influence of a radial offset on the globe visualization.

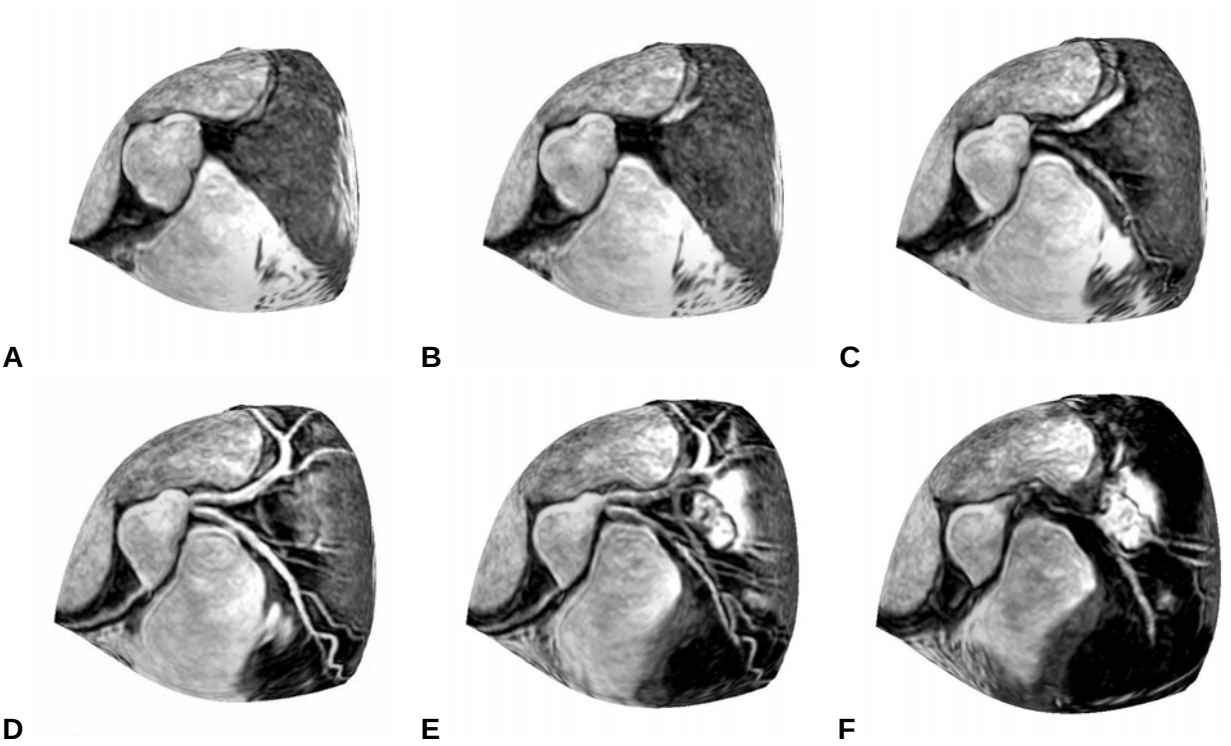


Figure 32: Radial offset for the globe surface. The depicted radial offsets are -6 mm (A), -4 mm (B), -2 mm (C), 0 mm (D), +2 mm (E), and +4 mm (F).

8.8.3.5 *Inflation / Deflation*

While the use of a radial offset grows and shrinks the surface everywhere, we would like to have the possibility to restrict the growing process to areas away from the coronaries. This way, the coronary tree is always visible, and growing and shrinking is similar to inflating and deflating a balloon where the defined vessel points act as a restricting force. Figure 34 shows an example with different inflation/deflation values.

The implementation concept is relatively simple. In a first step, a coarse interpolation and triangulation of the input points is done as described above, resulting in set of points lying on the globe surface called *inflation points*. We then update the inflation points by using the following algorithm:

```
FOR EACH inflation_point
   IF (inflation_point NOT CLOSE TO ANY vessel_point) THEN
      inflation_point.radius = INFLATE(inflation_point.radius, percentage)
```

```
   ENDIF
ENDFOR
```

The updated inflation points together with all the vessel points are then used in the subsequent surface computation as described above.

The current implementation of the `INFLATE` function as well as the criteria used in choosing the inflation points work well for small inflation/deflation values, but introduces artifacts when used in more extreme settings. The inflation/deflation option for the GlobeViz module is therefore just a prototype illuminating the general idea.

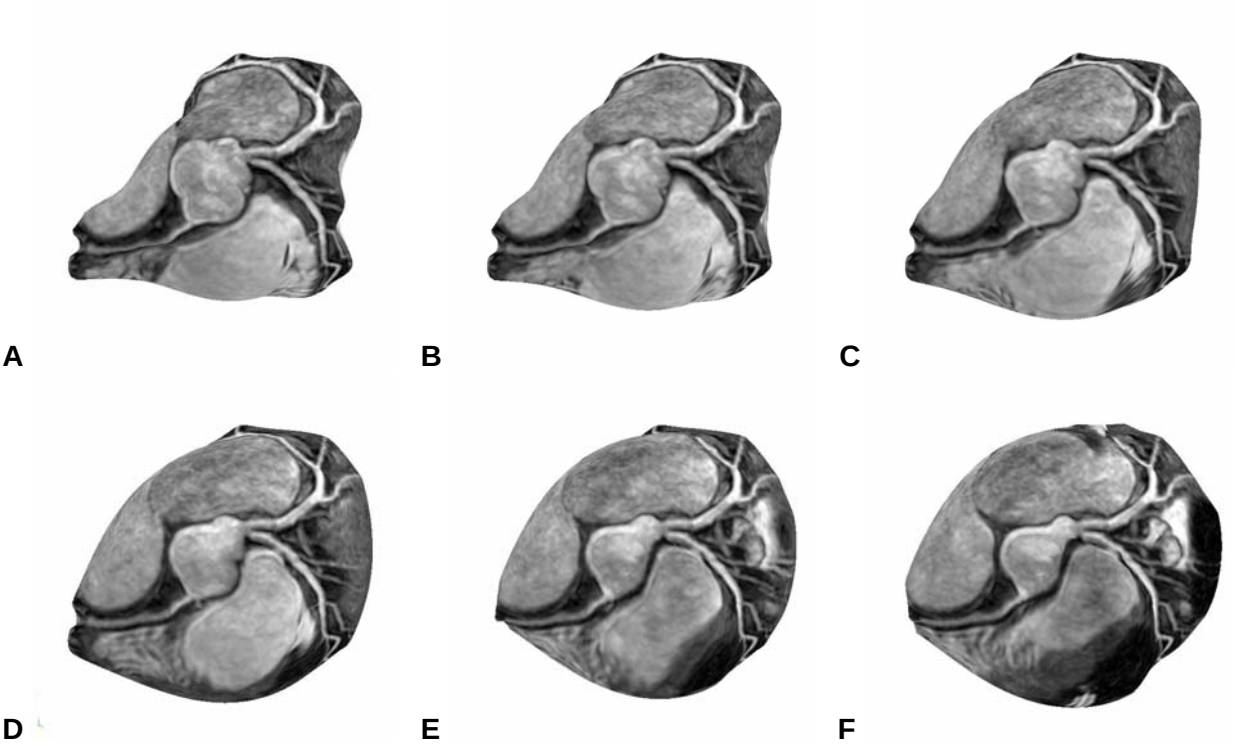


Figure 33: Deflation and inflaftion of the Globe surface. The deflation rate for (A) is -60%, for (B) -40%, and for (C)- 20%. (D) shows the original surface. (E,F) are inflated with a rate of 20% (E) and 40% (F). The vessel tree acts as a restriction to the inflation/deflation process and is visible in all images.

8.8.3.6 *Radial Thickness*

If the vessel is accurately defined with points all along the axis, a small radial thickness will optimize the vessel-to-background distinction. However, the complete vessel definition is either time consuming or relies on a very robust and accurate tracking algorithm. By increasing the radial thickness of the GlobeViz visualization, the vessels are visible even if there is only a subset of the points defined on the vessel axis. Figure 35 shows the influence of the radial MIP thickness in a case with only 6 points on the left circumflex defined.

The formula for creating the texture image by interpolation the original data is adapted to

$$surface[longrid, latgrid] = \max_{j=-s/2}^{s/2}\left(Interpolate(data, Position = longrid, latgrid, r(longrid, latgrid)) + j + offset\right)$$

where *s* is the radial thickness and *offset* the radial offset.

The option of a variable radial thickness allows the operator to see previously undefined side branches and extensions. Since it is possible to define new vessel points in the three-dimensional scene as well, the operator may start with a large thickness to detect new vessel points and then lower the thickness to optimize the vessel-to-background distinction. This procedure allows a fast and high-quality iterative assessment of the vessel under investigation.

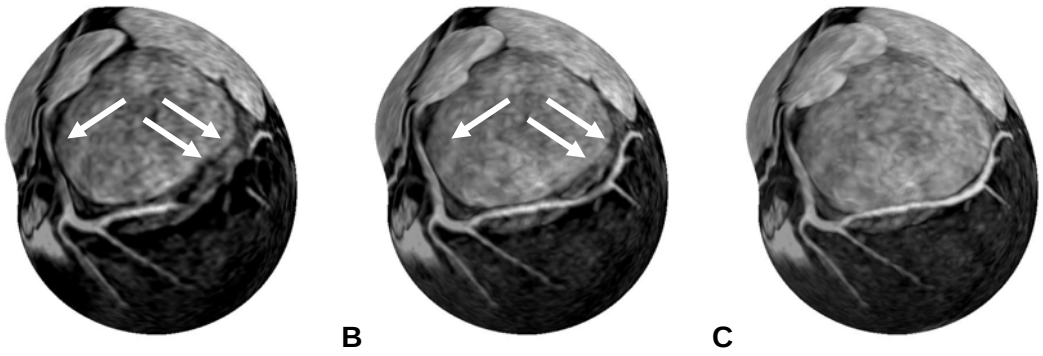


Figure 34: Influence of radial MIP thickness on GlobeViz visualizations. Only 6 points on the left circumflex are defined. (A) shows the resulting GlobeViz with a MIP thickness of 1 mm. The vessel does not appear connected (see arrows). In (B) and (C), a MIP thickness of 3 mm (B) and 6 mm (C) is selected and he vessel is continuously visualized.

8.8.3.7 *Data Source*

It may be useful to limit the set of used points to a sub-set of all vessels used for globe visualization procedure. Options include the use of points from the current vessel only or for any combination of $P_{i,LAD}$, $P_{i,RCA}$, and $P_{i,LCX}$. Figure 32 shows some sample combinations.

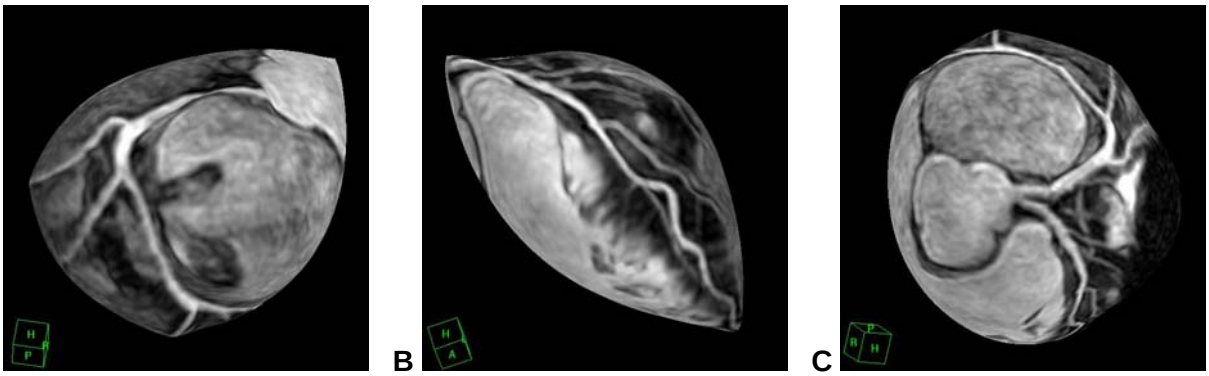


Figure 35: Restricted globe visualization with only a subset of all vessels. (A) depicts a globe visualization of the LCX only. (B) uses only the points defined on the LAD. In (C), points from LAD and LCX are used, but not the points defined on the RCA.

9 Masked Volume Projection ('TubeViz')

While the surface-based GlobeViz module succeeds in providing an overview of the whole coronary tree, a three-dimensional tool for the localized assessment of branches and stenosis is still missing. The vessel segmentation module is based on thresholding and has limited clinical significance because small parameter changes have a large impact on the visualization produced.

Maximum intensity projections are still considered the gold-standard in viewing MRA data. For coronary MRA as provided by the whole-heart sequence, MIP over slices or the whole data set do not work well because the arteries are over-projected by brighter organs surrounding the coronaries. We therefore introduce a module called 'TubeViz', which is based on masked volume projection. The mask consists of a tube with user-defined diameter along the vessels. The volume rendering function is chosen as a maximum intensity projection: the value of each pixel on the viewing plane is set to the brightest voxel, as determined by its opacity. The most opaque voxel's color appropriation is then reflected by the pixel on the viewing plane. Figure 36 shows the mask as well as the resulting projection and a region-growing segmentation in comparison.

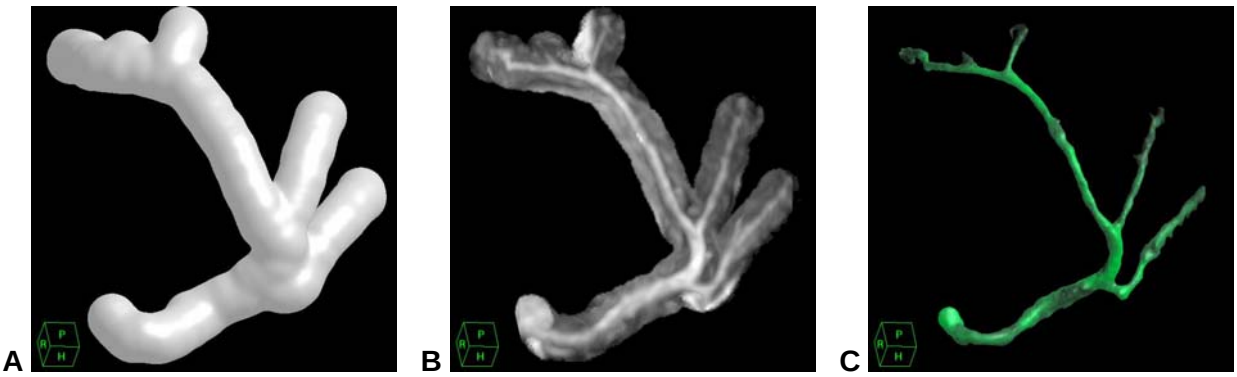


Figure 36: Masked volume projection ('TubeViz') of the left circumflex. The mask consists of a tube with user-defined diameter along the vessel (A). The tubular maximum intesity projection is performed with the original data (B), allowing an optimal assessment of the vessel structure without over-projection from other organs. The region-growing segmented vessel is shown in comparison (C).

The choice of maximum intensity instead of alpha-rendering for the projection results in images very similar to the one achieved with MPR or curved reformatting, however they are truly three-dimensional. Therefore, the operator can pan, zoom and rotate the data to navigate into an optimal view position. The contrast level and center (window) are identical throughout the program and can be interactively adapted. Figure 37 shows the use of TubeViz visualization for the local assessment of a branching left circumflex coronary artery.

9.1.1 Options for the masked volume projection

Options for the *TubeViz* module are the maximum depicted vessel radius and a choice of the interpolation method used.

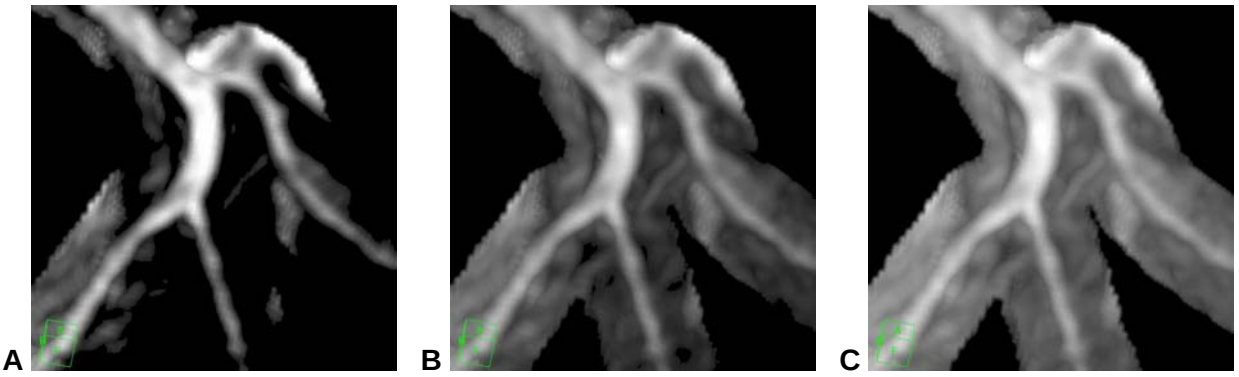


Figure 37: Interactive windowing allows the adaption of contrast level and center to enhance the visualized volume of interest.

9.1.1.1 *Tube Radius*

The tube radius is the only parameter input for the TubeViz visualization. Over-projection from other organs is minimized by choosing a small radius. On the other side, a large radius allows the visualization of branching vessel parts even if they have not been marked with vessel points. Therefore, the tube radius is usually a trade-off between coverage and visibility. Alternatively, a large value may initially be used to find branching vessels, and then a smaller value is applied to assess the found vessel structure. Figure 38 presents the tube visualization with different tube radii together with the mask used in calculating the MIP.

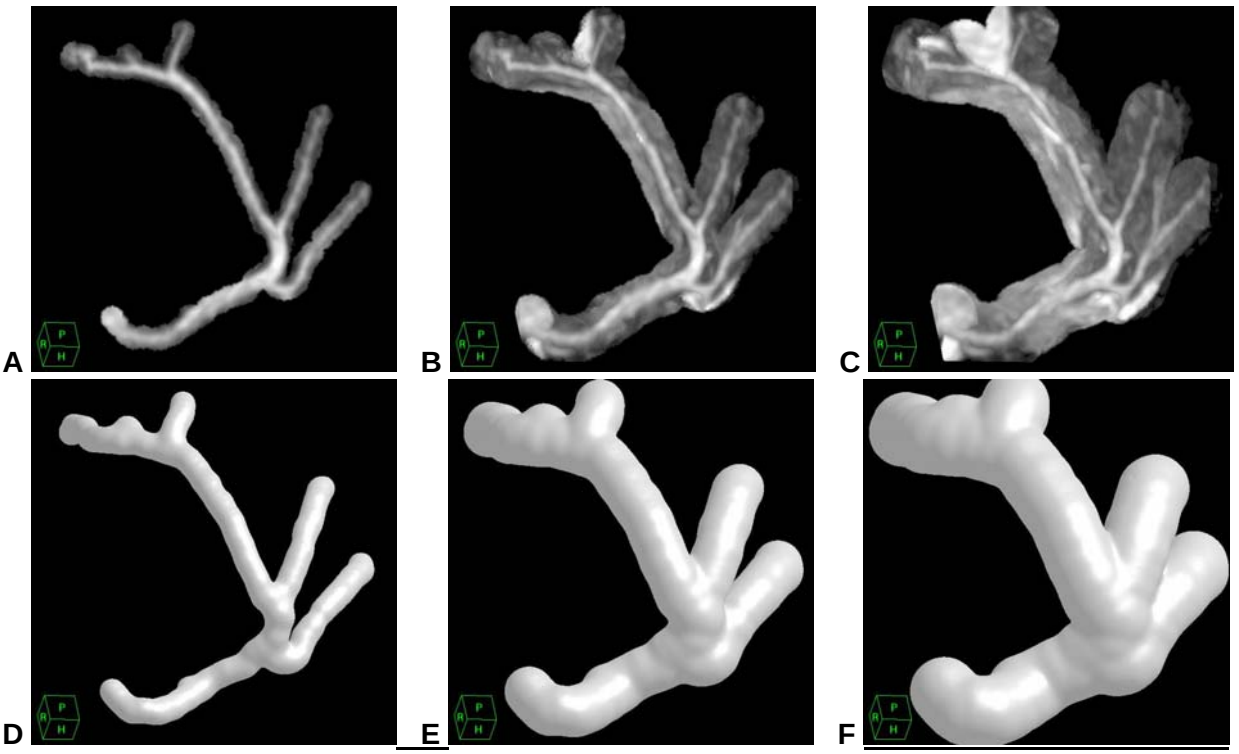


Figure 38: Different tube radii for the masked volume projection visualization. (A) = 2.5 mm, (B) =5 mm, (C) = 7.5 mm. (D,E,F) show the volume used when computing the MIP projections.

9.1.1.2 *Interpolation method*

It's possible to use either *trilinear interpolation* or *nearest neighbor sampling* to determine the data value for each step on a ray. Trilinear interpolation improves the quality of the images produced but is much more expensive, especially when the volume has low resolution with respect to the size of the viewing plane. Figure 39 shows a comparison of the two different settings.

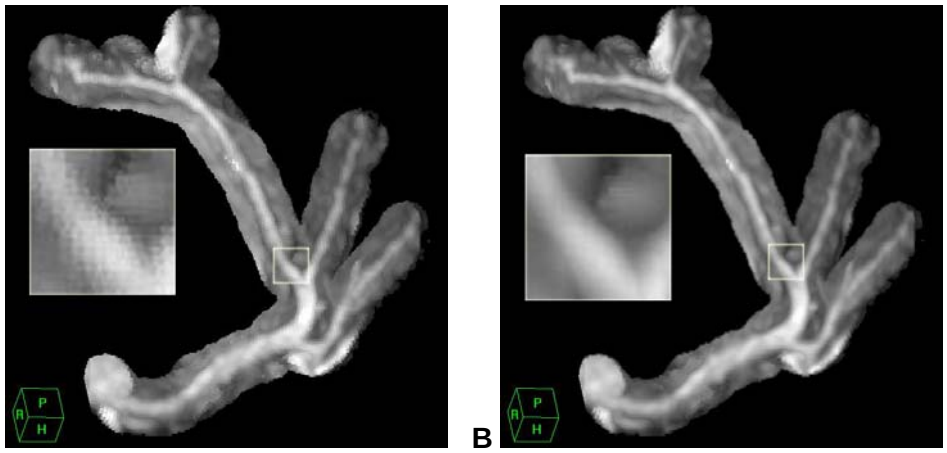


Figure 39: Trilinear interpolation provides better image quality, but results in a slower volume MIP. In (A), nearest neighbor sampling is used, whereas (B) employs trilinear interpolation.

9.2 Quantification and Measurements

Quantification of MRA data is important for both research and clinical use. In research, a quantitative comparison between different scanning methods and modalities is needed to objectively assess new methods. In a clinical setting, position and length measurements are important e.g. for an indication of stent length and placement. Also, a stenosis grading based on diameter measurement is often used as an aid to decide between different treatment options.

While vessel length quantification can be implemented with little effort, diameter and cross-section measurements are much more complex and require knowledge about the imaging method involved. There are basically four types of measurement techniques: densitometric, derivative based, threshold-based, and model-based. The proper and meaningful application of each approach requires a thorough understanding of the limitations involved. An overview can be found in (Hoffmann, Nazareth et al. 2002). In (de Koning, Schaap et al. 2003), a vessel model with a *non-uniform rational B-spline* (NURBS) surface is deformed and matched to the underlying data. This procedure ensures a smooth vessel wall modeling, but is critical for the application on pathological data.

9.2.1 Implementation

Length measurement is based on a user-defined selection of vessel points. The operator first highlights points for the wanted vessel length and then presses the update button. The length of a natural spline fitted through the points is taken as the approximation of the vessel length. The user has full control over the points used in defining the spline, and visual three-dimensional feedback on the distance measured is provided.

9.2.2 Diameter and Area

The implemented diameter and area quantification module relies on the knowledge of the vessel centerline as defined by the tracking algorithm. The problems of vessel measurement and vessel segmentation are closely related – in both cased would we like to distinguish between image parts within and outside the vessel.

In a first step, vessel lumen segmentation on the cross-sectional image is obtained at the point of investigation, using an algorithm similar to the one used in the segmentation module. Instead of relying on a fixed percentage value for the *full with at % maximum criterion*, the operator can adapt the percentage to a value that provides adequate lumen segmentation for the image modality and method used. For the case of whole-heart data as provided by (Weber, Martin et al. 2003) we achieved the best results with a threshold value of 80 % (corresponding to a full width at 80% maximum criterion).

Secondly, a radial diameter sweep is performed from the brightest point in the cross-sectional image to the detected borderline. The minimum, maximum, and average diameter is then recorded and the quantification procedure repeated for each user-defined vessel point.

Figure 30 shows the different steps for the CoroViz vessel quantification. It is important to note that the current diameter and area measurement implementation is based solely on the coarse cross-section image obtained and fails frequently for distal parts and / or when other bright structures are near.

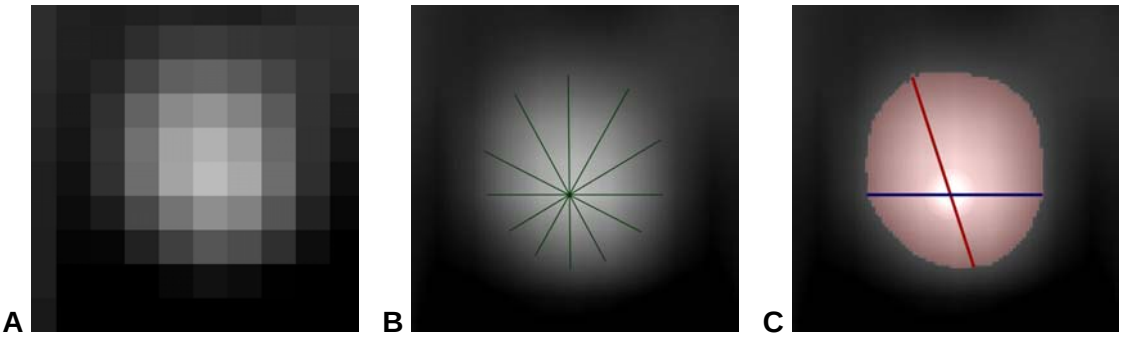


Figure 40: CoroViz vessel quantification. The cross-section image (A) perpendicular to the vessel direction is interpolated. A diameter sweep is performed from the brightest point in the interpolated image with the full-width at 80% center value criterion used for border detection (B). The minimum, maximum and average diameter are recorded as well as the total area inside the border (highlighted, C).

9.3 Output

The output options of the CoroViz-Software are adapted to the current use in research. Therefore, there is no printing function currently implemented. However, the integration of even an advanced reporting tool with images, cross-section data and stenosis quantification would be unproblematic. The current output functions are aimed to allow an easy incorporation into the research workflow.

9.3.1 Images

All the two-dimensional reformats as well as screenshots of the three-dimensional scene can be saved to standard JPEG, TIFF or PNG formats. Also, the data can be copied to the clipboard for later use e.g. in Microsoft Word or PowerPoint.

9.3.2 Movies

An intuitive movie tool allows for fast generation of animations and movies to catch the three-dimensional scene. Sample movies can be found on the accompanying CD-ROM.

9.3.3 Vessel point definitions

Sets of points defining the coronary anatomy can be saved to and restored from file. These sets can only be used within the CoroViz software, but they can be loaded on different dataset e.g. for the comparison of different scanning sequences.

9.3.4 Filtered data caching

The vessel-enhanced filtered data can be saved to and restored from file to facilitate frequent processing of the same dataset.

9.3.5 Measurements

The measured vessel length is displayed numerically. The information about vessel minimum, maximum and average diameter as well as the cross-sectional area can be copied and easily be imported into an Excel worksheet.

10 Results

9 datasets were successfully processed. The time required to automatically track the vessel centerline and generate the different visualization was <3 minutes for all datasets; a manual refinement and alignment of the vessel axis for optimal results added between 1 and 15 minutes to the preparation time.

Figure 41 shows a screenshot of the running CoroViz Software. The current vessel under investigation is the right coronary artery. An axial, coronal, and sagittal view are shown permanently in the upper three viewports. The points found by the tracking algorithm are displayed with crosses. The current selected two-dimensional reformat is shown in the lower left viewport, in this case a curved reformat. In the lower right viewport, the cross-sectional image of the current position is permanently shown. The option tabs for the different visualization modules is arranged between the lower two viewports.

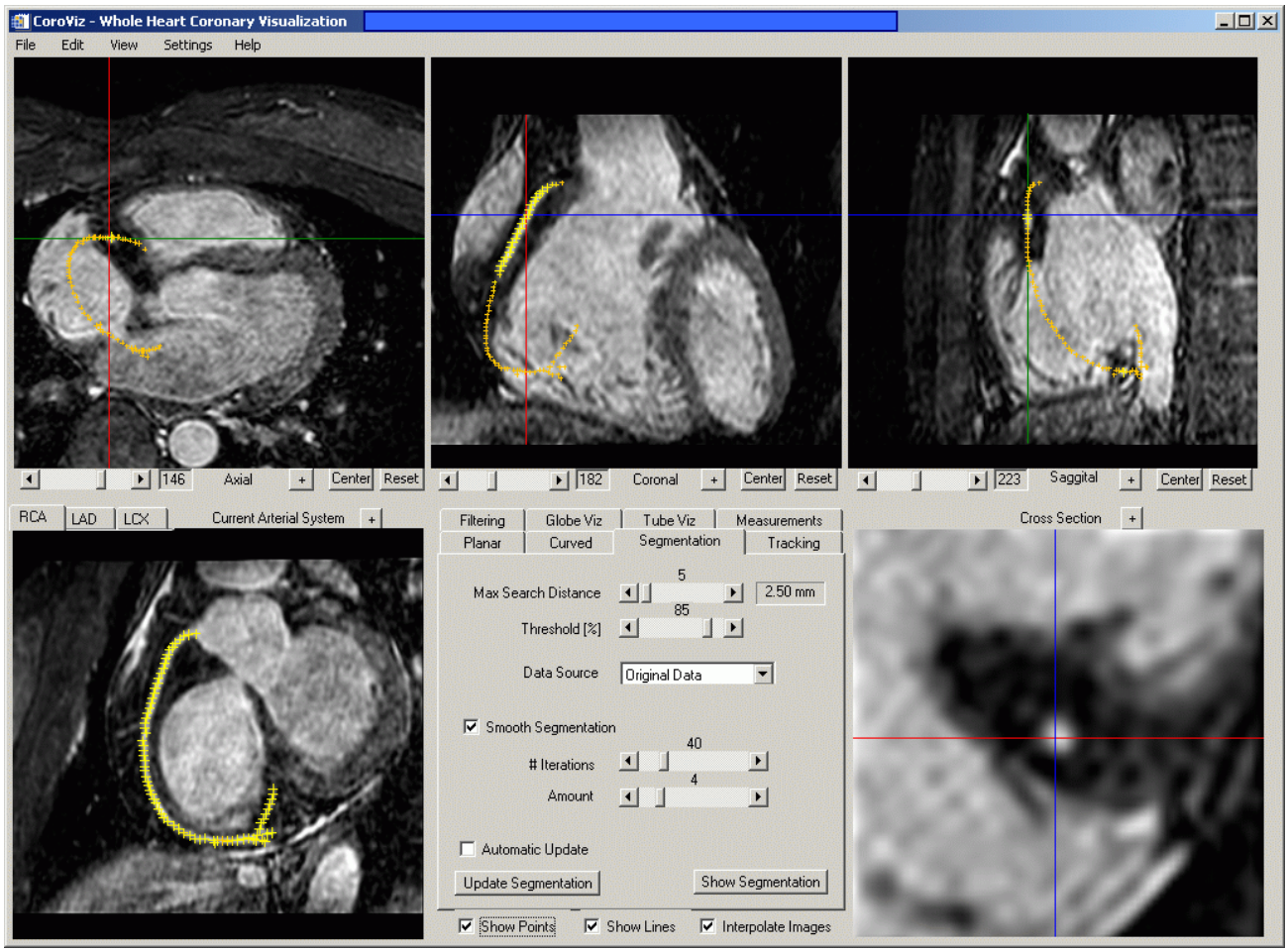


Figure 41: Screenshot CoroViz Software

Figure 42 displays a screenshot of the CoroViz 3D Visualizer. The orientation cube in the lower left corner shows the current view direction. The transparency of the different visualization modules can be interactively adapted. 8 standard views for the left system and 3 standard views for the right system provide a fast overview of the coronary tree. An intuitive movie recorder allows the fast generation of animations.

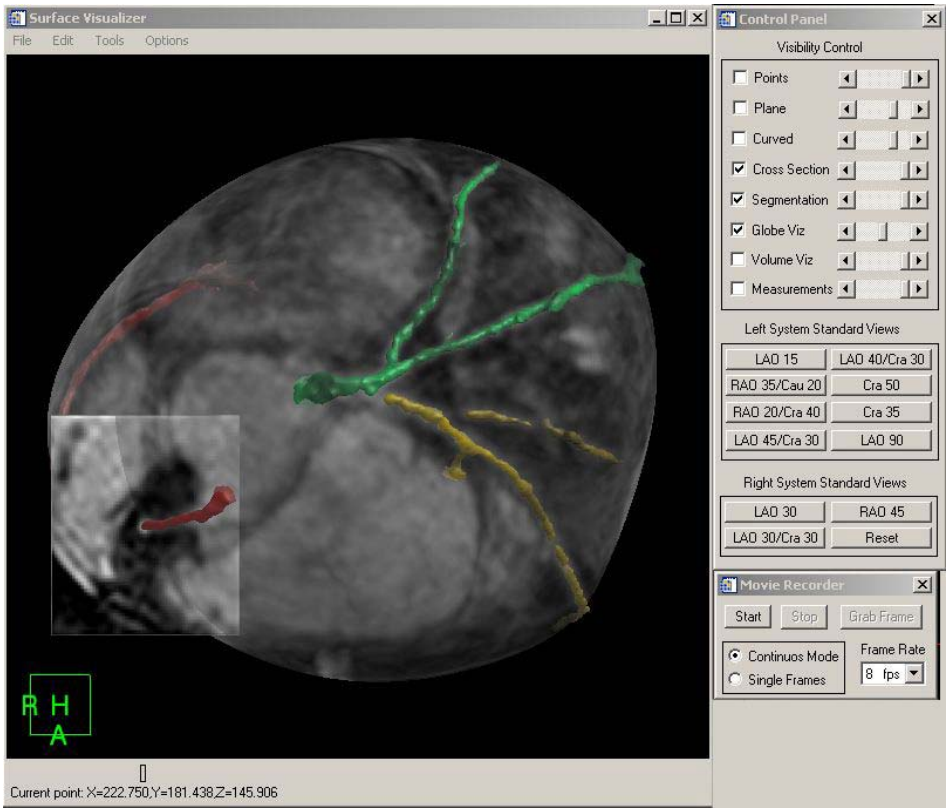


Figure 42: Screenshot CoroViz 3D Visualizer.

10.1 Tracking

For a clinical use it is interesting to know how successful the algorithm is in a more automated setting with just one point defined on each vessel. Table 3 shows the results for 2 different datasets. One point was defined on the RCA, the LAD, and the LCX about 30 mm from the vessel origin. The vessel tracking module was then run on the filtered data with a 30 % threshold setting. *Found length* is the length of the tracked vessel that is congruent with the manually defined vessel axis. *Erroneous length* is the sum of the length of all tracked branches that are not part of the vessel structure and would have to be removed for further processing. *Manually defined* corresponds to the length of the manually tracked vessel.

Table 3: Results of the CoroViz Tracking algorithm for 2 different data sets.

Source Data	*Branch*	*found length*	*erroneous length*	*Manually defined*
1	RCA	118 mm	0 mm	121 mm
1	LAD	52 mm	12 mm	86 mm
1	LCX	32 mm	0 mm	84 mm
2	RCA	41 mm	0 mm	59 mm
2	LAD	120 mm	21 mm	141 mm
2	LCX	34 mm	40 mm	125 mm

The success rate of the semi-automatic tracking algorithm is highly dependant on the quality of the source data and the absence of external bright structures close to the vessel axis. For the first dataset, the RCA was almost completely found, but results for the left system were poor. The LAD of the second dataset was detected to 85 %, yet artificial side branches were included in the vessel tree as well. Results achieved for RCA and LCX were unsatisfactory.

However, the tracking module can be also iteratively used in a more interactive way and is very helpful in detecting additional vessel branches. Use of the vessel tracking module in a assisted way reduces the time needed to define the RCA from around 3 minutes to less than one minute and for the left system from about 15 minutes to fewer than 5 minutes for an experienced operator.

10.2 2D Visualization

Two-dimensional visualizations include the cross-sectional images, multi-planar reformat (MPR) with minimum squared-distance to the vessel, and curved reformats similar to the ones provided by the Soap-Bubble tool (Etienne, Botnar et al. 2002). Data source for the curved reformats can be either the original data or the MPR image stack. Figure 43 shows three curved reformats from the RCA, LCX, and LAD respectively.

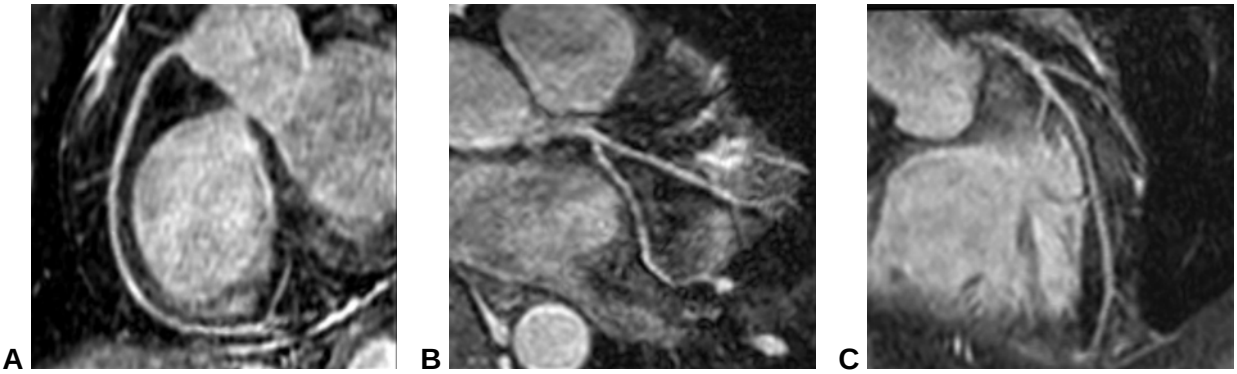


Figure 43: Curved reformats for the individual vessels. All reformats are from the same data set. (A) RCA, (B) LCX, (C) LAD

10.3 3D Visualization

Three-dimensional visualization options include segmentation display, globe visualization ('GlobeViz'), and masked volume MIP ('TubeViz'). Moreover, the two-dimensional modules (multi-planar and curved reformats, cross-sectional image) can be blended into the scene, allowing three-dimensional inspection and control.

Figure 44-A shows a globe visualization of all the coronary vessels. The deformed sphere can be interactively rotated, zoomed, panned, and windowed. Figure 44-B presents the segmented coronaries, together with the globe visualization (set to a 40% transparency value), allowing simultaneous perception of the coronaries and the shape of the deformed surface. Figure 44-C and 44-D show the left circumflex coronary artery. A three-dimensional curved reformat (MIP slice thickness 1.5 mm) and the vessel centerline found by the tracking algorithm are displayed in Figure 44-C the masked volume projection module restricts the MIP to a tubular structure around the vessel center line, allowing an accurate local assessment of the vessel structure (Figure 44-D).

Figure 45 shows an example of iterative vessel point definition. Figure 45-A depicts the globe visualization of vessel points found in the tracking process. Figure 45-B displays tube visualization with a tube radius of 3 mm. In Figure45-C, the tube is enlarged to 5 mm, and the beginning of an additional side branch close to the origin of the RCA is visible (see arrows). The operator then defines supplementary vessel points with a mouse click on the three-dimensional globe. The tube visualization is automatically updated and shows the updated RCA (Figure 45-D).

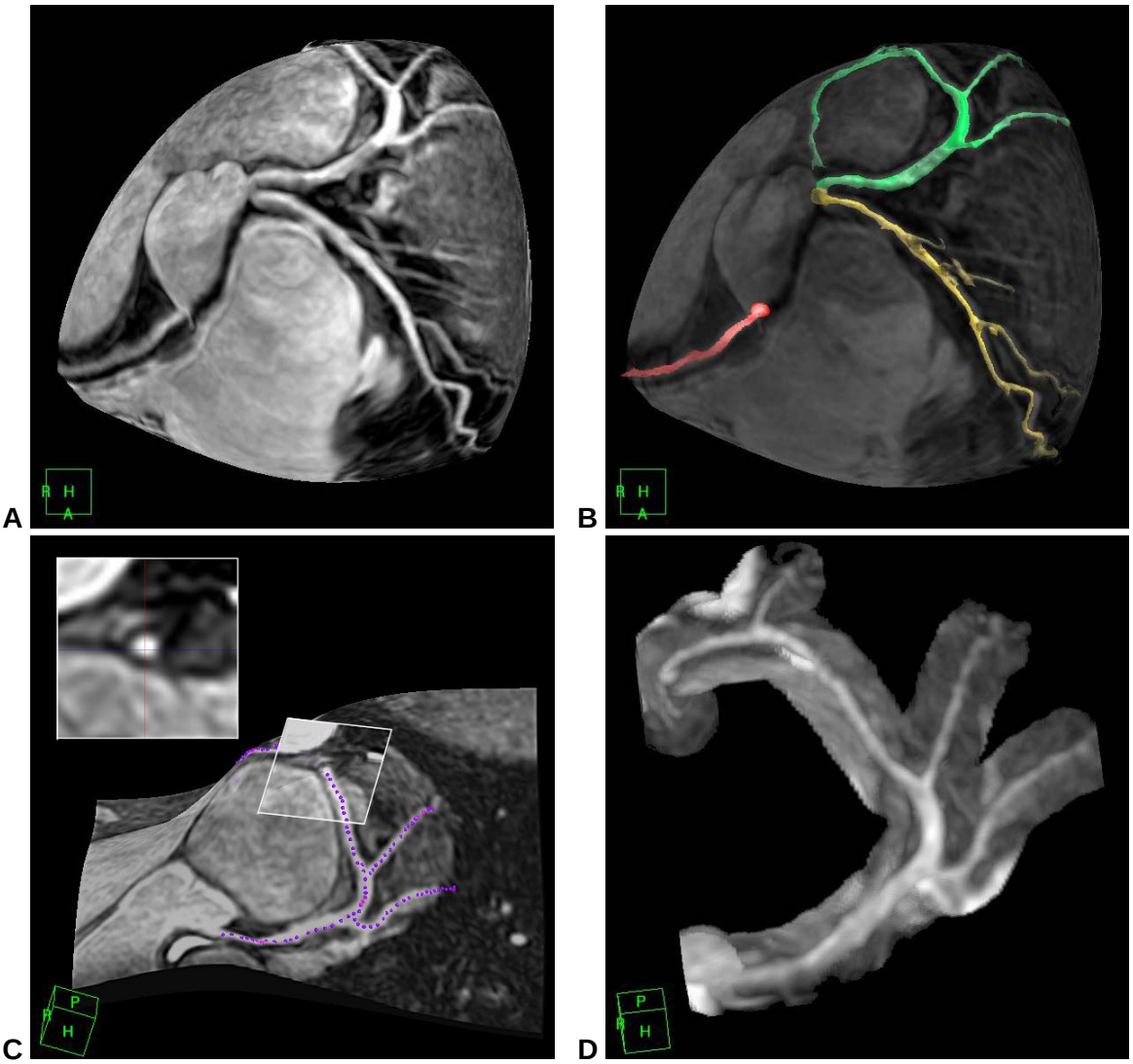


Figure 44:Three-dimensional CoroViz Visualization. (A) shows a globe visualization of all the coronary vessels. (B) depicts the segmented coronaries, together with a transparent globe visualization, allowing simultaneous perception of the coronaries and the shape of the deformed surface. (C,D) show the left circumflex coronary artery. (C) displays a curved reformat and the vessel centerline found by the tracking algorithm. A tube visualization of the same vessel is shown in (D).

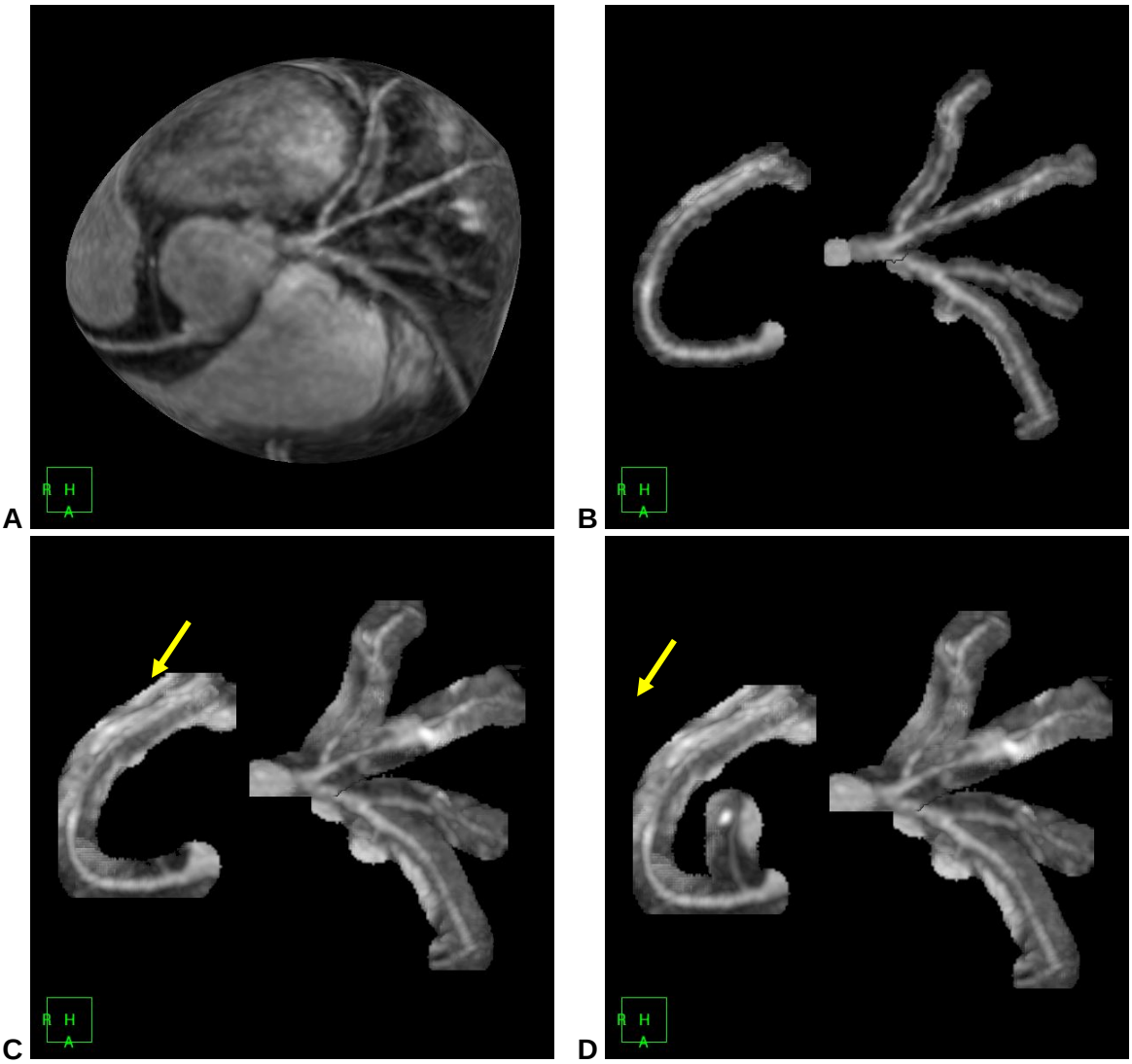


Figure 45: 3D-Visualization and iterative vessel point definition. (A) shows a globe visualization. (B) displays a tube visualization with a tube radius of 3 mm, which is enlarged to 5 mm in (C). The beginning of an additional side branch close to the origin of the RCA is visible (see arrows). Supplementary vessel points are defined on the three-dimensional globe, and the tube visualization is automatically updated, displaying the newly found branch (D).

10.4 Length Measurements

Table 4 provides a comparison of measurements on the coronary tree performed with the *VirtualPlace Advance* software, the *Soap-Bubble* tool, and the *CoroViz* software package. Aim of the study was a verification of the length measurement module and not a comparison of visualization techniques. Therefore, the operator tried to find the same branching and end points in each tool. The numbers in the *Branch* column correspond to Figure 46.

Table 4: Comparison of length measurements with the Soap-Bubble tool, VirtualPlace Advance, and the CoroViz package.

Branch	*Soap-Bubble*	*VirtualPlace Advance*	*CoroViz*
1-2-3	150.1 mm	143.2 mm	138.4 mm
1-2-4	92.0 mm	93.8 mm	96.7 mm
1-5-6	91.4 mm	88.7 mm	93.7 mm
1-5-7	142.8 mm	151.5 mm	140.2 mm
8-9	64.3 mm	62.6 mm	63.8 mm

The resulting lengths are similar for the three programs. However, it was very time consuming to find the identical points in the different software packages. In general, measurements with the CoroViz software package were obtained in shortest time.

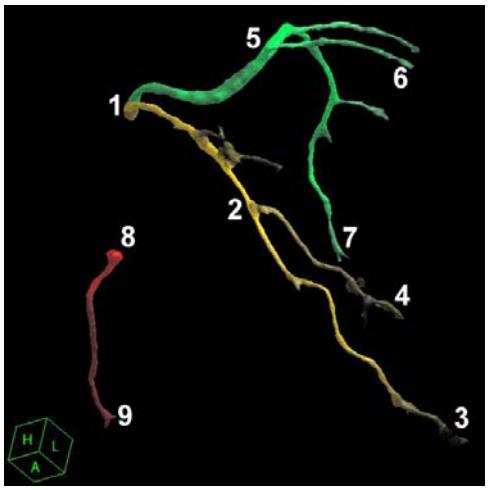


Figure 46: Branch labeling for lenght measurement.

10.5 Diameter and Area Measurements

We demonstrate success and failure of diameter and area measurements on a 60 mm part of the LAD. Figure 47 shows the exact places where measurements are taken along with the starting point [0 mm] and end point [60 mm]. Three representative measurements at positions (1) [7.2 mm], (2) [32.1 mm], and (3) [48.3 mm] are presented in Figure 49. Measurement (1) was taken from a part where the lumen is a very symmetric blob, causing little difficulties for the measurement algorithm. (2) is taken from a position where another vessel branches off. The min/max diameters indicate the amount of skewness in the lumen. In (3), the lumen segmentation leaks out and corrupts the measurement. Figure 48 displays a graphical representation of the entire measuring section.

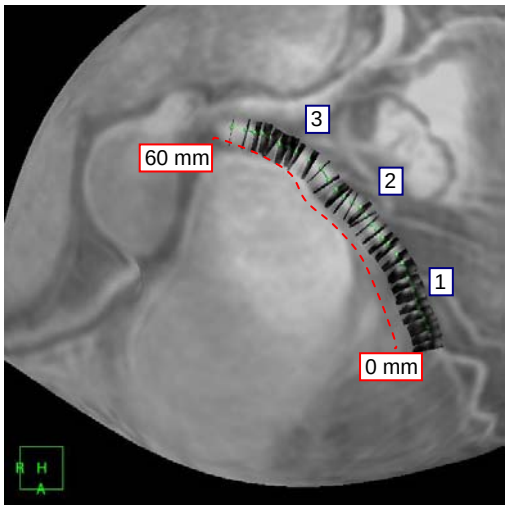


Figure 47: Placement of diameter and area measurements on the LAD.

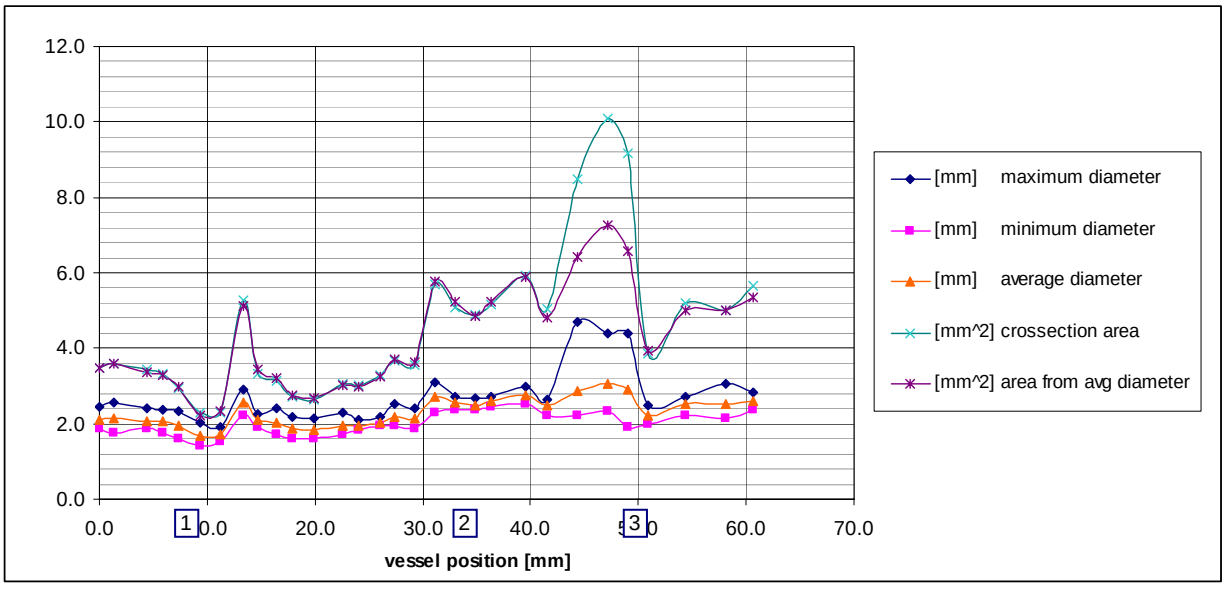


Figure 48: Two-dimensional vessel quantification. The vessel position is related to Figure 47. The position of the measurements (1,2,3) are marked on the position axis.

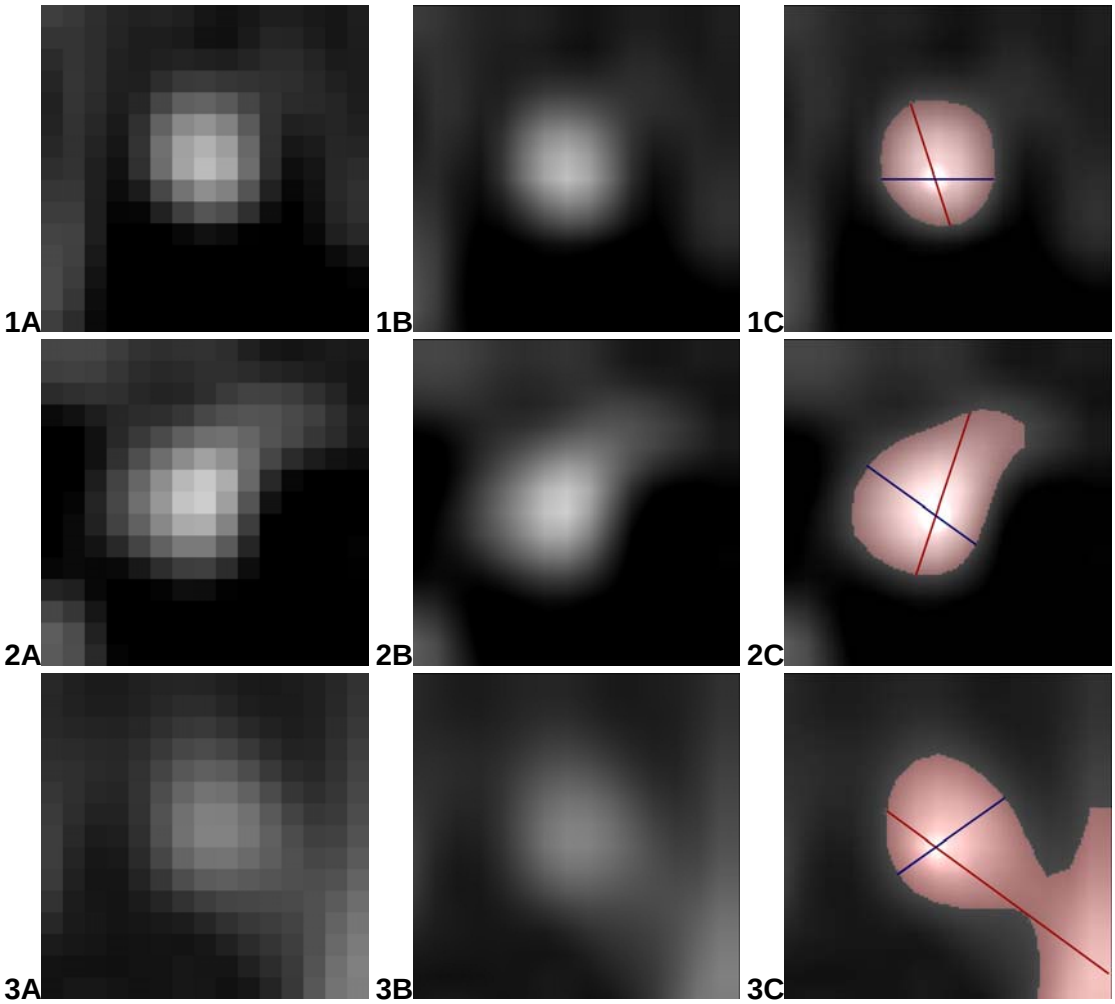


Figure 49: Comparison of results achieved with the CoroViz quantification tool. (A) shows the cross-section perpendicular to the vessel direction which is interpolated for further processing (B).The minimum, maximum and average diameter are recorded as well as the total area inside the border (highlighted, C). Measurement (1) was taken from a part where the lumen is a very symmetric blob, causing no problems for the measurement process. (2) is taken from a position where another vessel branches off. The min/max diameters indicate the amount of skewness of the lumen. In (C), the lumen segmentation leaks out and corrupts the measurement.

11 Discussion

The current implementation of the CoroViz user interface allows a fine-tuning of most settings, even individually for the different vessels. This permits an easy adoption of the software to various uses in a research environment which cannot be predicted yet. Implementing the ideas for clinical use would require a pre-configuration for different data acquiring methods and then a reduction to a much simpler user interface.

11.1 Globe Flattening

Even though the globe visualization allows interactive rotation, it is often desirable to have a two-dimensional ('flat') image with all the information included for further processing. In the derivation of the globe visualization we frequently reasoned by analogy with geodesy. The goal of map projection is to depict the three-dimensional earth globe on a two-dimensional image. Various projection types such as *polar*, *stereographic*, *azimuthal*, or *gnomonic* are used depending on the requirements of the application. Very specific projections exist for the optimal depiction of the five continents. However, those projections are not very suitable for the flattening of the coronary globe. Figure 50 shows *interrupted Goode* and *Mollweide* projections for both the earth and coronary globe. For clinical routine processing it would be helpful to develop an optimal flattening technique which minimizes discontinuations and distortions of the coronary arteries.

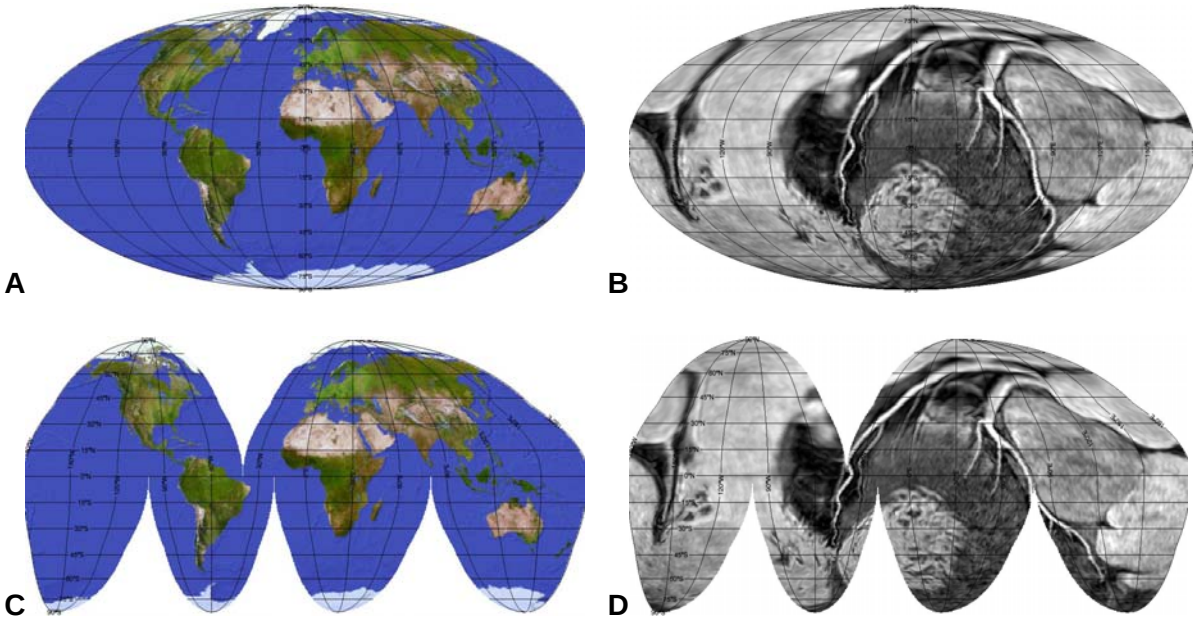


Figure 50: Analogy between earth map projection and globe flattening. (A,B) show a Mollweide projection for both the earth and the coronary globe. Interrupted Goode projections are shown in (C) and (D).

11.2 Tracking algorithm

Even though the tracking module reduces the tedious work of the operator in manually defining the vessel tree, the tracking algorithm would have to be refined to allow usage in an automated environment.

11.3 Missing symbolic vessel tree

Unfortunately, there is no symbolic vessel tree present from which we could easily generate the mask for the tube visualization. The points are not structured, so there is no explicit information on the position of branching or end points available. The center-line is simply a set of points that happen to be pair-wise close together. The current mask is therefore simply a union of all spheres with the maximal tube value as radius and the points as center. If the vessel points are close together (e.g. as provided by the tracking algorithm), this arrangement resembles a tube. However, if there are gaps or the vessel center-line is not fully defined, then the used spheres are visible. Figure 51 shows an example. This limitation similarly affects the segmentation tool. It is important to note that this problem is not associated with the general TubeViz idea and could be solved by generating the symbolic vessel tree during the tracking process.

Figure 51: Tube visualization with incomplete vessel center-line. The current implementation simply uses the union of spheres with center at the defined points as a mask (A). Therefore, the resulting tube visualization contains gaps where no points are defined on the vessel (B). The TubeViz with all points defined is shown in (C).

12 Conclusions

A comprehensive tool for two- and three-dimensional visualization and quantification of whole-heart coronary MRA has been developed. Globe and tube visualization techniques allow the three-dimensional assessment of the coronary arteries with only marginal concealment from signals originating outside the vessel.

All visualization techniques stay close to the original source and the different modules are depicted in the same three-dimensional scene, allowing a comparison and assessment of the spatial composition. The scene can be interactively zoomed, rotated, panned, and windowed.

The look-and-feel is made similar to standard software packages such as Gyroview or EasyVison, therefore users with general scanner software experience will feel familiar with the graphical user interface. The use for clinical routine image post-processing and modalities other than coronary MRA remain to be investigated.

13 References

Anderson (1999). "LAPACK Users' Guide." SIAM 3rd edition.

Boer, R. W. d. (2000). "A new family of MR systems: Gyroscan Intera - System design and clinical applications." medicamundi 44(1).

Bülow, T., C. Lorenz, et al. (2003). A General Framework for Tree Segmentation and Reconstruction from Medical Volume Data. Hamburg, Germany, Philips Research Laboratories.

de Koning, P. J., J. A. Schaap, et al. (2003). "Automated segmentation and analysis of vascular structures in magnetic resonance angiographic images." Magn Reson Med 50(6): 1189-98.

Deshpande, V. S., S. M. Shea, et al. (2001). "3D magnetization-prepared true-FISP: a new technique for imaging coronary arteries." Magn Reson Med 46(3): 494-502.

Dijkstra, E. (1959). "A note on two problems in connection with graphs." Numerische Mathematik(1): 269-271.

Earls, J. P., V. B. Ho, et al. (2002). "Cardiac MRI: recent progress and continued challenges." J Magn Reson Imaging 16(2): 111-27.

Etienne, A. (2001). Quantitative Coronary Analysis Based on Three Dimensional Coronary Magnetic Resonance Angiography. Physics. Zurich, Swiss Federal Institute. Diplomarbeit.

Etienne, A., R. M. Botnar, et al. (2002). ""Soap-Bubble" visualization and quantitative analysis of 3D coronary magnetic resonance angiograms." Magn Reson Med 48(4): 658-66.

Falcao, A. X. and J. K. Udupa (2000). "A 3D generalization of user-steered live-wire segmentation." Med Image Anal 4(4): 389-402.

Felkel, P. and R. Wegenkittl Vessel Tracking in Peripheral CTA Datasets – an overview. Vienna Austria, VRVis Center, Austria, felkel@vrvis.at.

Frangi, A., W. Niessen, et al. (1998). "Multiscale Vessel Enhancement Filtering." Lecture Notes in Computer Science 1496.

Frangi, A. F., W. J. Niessen, et al. (1999). "Model-based quantitation of 3-D magnetic resonance angiographic images." IEEE Trans Med Imaging 18(10): 946-56.

Hoffmann, K. R., D. P. Nazareth, et al. (2002). "Vessel size measurements in angiograms: a comparison of techniques." Med Phys 29(7): 1622-33.

Kanitsar, A., D. Fleischmann, et al. CPR - Curved Planar Reformation. Vienna Austria, Institute of Computer Graphics and Algorithms, Vienna University of Technology University.

Lakare, S. (2000). 3D Segmentation Techniques for Medical Volumes. Department of Computer Science. New York, State University of New York at Stony Brook. Research Proficiency Exam.

Lorenz, C. (1997). "Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2D and 3D medical images." Proc. CVRMed-MRCAS'97: 233-242.

McCarthy, R. M., S. M. Shea, et al. (2003). "Coronary MR angiography: true FISP imaging improved by prolonging breath holds with preoxygenation in healthy volunteers." Radiology 227(1): 283-8.

Olabarriaga, S. D., M. Breeuwer, et al. (2003). "Minimum Cost Path Algorithm for Coronary Artery Central Axis Tracking in CT images." Medical Image Computing and Computer-Assisted Intervention 2879: 687-694.

Sato, Y., S. Nakajima, et al. (1998). "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images." Med Image Anal 2(2): 143-68.

Shea, S. M., V. S. Deshpande, et al. (2002). "Three-dimensional true-FISP imaging of the coronary arteries: improved contrast with T2-preparation." J Magn Reson Imaging 15(5): 597-602.

Udupa, J. K. (1999). "Three-dimensional visualization and analysis methodologies: a current perspective." Radiographics 19(3): 783-806.

van Bemmel, C. M., M. A. Viergever, et al. (2004). "Semiautomatic segmentation and stenosis quantification of 3D contrast-enhanced MR angiograms of the internal carotid artery." Magn Reson Med 51(4): 753-60.

van Bemmel, C. M., O. Wink, et al. (2003). "Blood pool contrast-enhanced MRA: improved arterial visualization in the steady state." IEEE Trans Med Imaging 22(5): 645-52.

Weber, O. M., A. J. Martin, et al. (2003). "Whole-heart steady-state free precession coronary artery magnetic resonance angiography." Magn Reson Med 50(6): 1223-8.

Westenberg, J. J., R. J. van der Geest, et al. (2000). "Vessel diameter measurements in gadolinium contrast-enhanced three-dimensional MRA of peripheral arteries." Magn Reson Imaging 18(1): 13-22.

William E. Lorensen, H. E. C. (1987). "Marching Cubes: A High Resolu-tion 3D Surface Construction Algorithm." Computer Graphics 21(4): 163-169.

William H. Press, S. A. T., Brian P. Flannert, William T. Vetterling (1988). Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press.

Wink, O., A. F. Frangi, et al. (2002). "3D MRA coronary axis determination using a minimum cost path approach." Magn Reson Med 47(6): 1169-75.

Wink, O., W. J. Niessen, et al. (2000). "Fast delineation and visualization of vessels in 3-D angiographic images." IEEE Trans Med Imaging 19(4): 337-46.

Wink, O., W. J. Niessen, et al. (2004). "Multiscale vessel tracking." IEEE Trans Med Imaging 23(1): 130-3.

Yasutaka Ichikawa, H. S., Naohisa Suzawa, Mikinori Nagata, Katsutoshi and T. H. Makino, Kan Takeda, Nozomu Koyama, Marc van Cauteren (2004). "Clinical Feasibility of Whole Heart Coronary MRA using a Navigator-gated 3D bTFE

Sequence." Proceedings ISMRM Kyoto 704.

Yi, J. and J. B. Ra (2003). "A Locally Adaptive Region Growing Algorithm for Vascular Segmentation." International Journal of Imaging Systems and Technology 13(4).

Zahlten, C. (1995). Beiträge zur mathematischen Analyse medizinischer Bild- und Volumendatensätze. Bremen, Universät Bremen. PhD thesis.

14 Appendix

14.1 Plane Fitting Algorithm

The points with coordinates *(x,y,z)* that lie on a plane fulfill the equation

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \qquad (1)$$

where the vector *(a,b,c)* is the plane normal and $a^2 + b^2 + c^2 = 1$. If the number of predefined points is smaller than three, the plane is not uniquely defined. For three points, the solution of the equation is trivial. If the number of points exceeds three, we fit the plane by minimizing the sum of the squared distance from all the points.

The distance R_i of a point *i* from the plane is

$$R_i = a \cdot x_i + b \cdot y_i + c \cdot z_i - d \qquad (2)$$

Given a set of *n* points x_i, y_i, z_i we wish to minimize the sum of the distances

$$Q = \sum_{i=1}^{n} R_i = \sum_{i=1}^{n} \left(a \cdot x_i + b \cdot y_i + c \cdot z_i - d\right)^2 \qquad (3)$$

To minimize, we set the first derivative equal zero

$$\frac{dQ}{da} = \sum_{i=1}^{n} 2 \cdot x_i \left(a \cdot x_i + b \cdot y_i + c \cdot z_i - d\right)^2 = 0 \qquad (4)$$

$$\frac{dQ}{db} = \sum_{i=1}^{n} 2 \cdot y_i \left(a \cdot x_i + b \cdot y_i + c \cdot z_i - d\right)^2 = 0 \qquad (5)$$

$$\frac{dQ}{dc} = \sum_{i=1}^{n} 2 \cdot z_i \left(a \cdot x_i + b \cdot y_i + c \cdot z_i - d\right)^2 = 0 \qquad (6)$$

$$\frac{dQ}{dd} = \sum_{i=1}^{n} -2 \cdot \left(a \cdot x_i + b \cdot y_i + c \cdot z_i - d\right)^2 = 0 \qquad (7)$$

Equation (7) can be rewritten as

$$d = a \cdot x_0 + b \cdot y_0 + c \cdot z_0 \qquad (8)$$

with $x_0 = \sum_{i=1}^{n} x_i / n$, $y_0 = \sum_{i=1}^{n} y_i / n$, and $z_0 = \sum_{i=1}^{n} z_i / n$

Equation (8) shows that the best plane passes through the centre of mass. Subtracting the centre of mass from each point and substituting into (4-6) we get a set of simultaneous equations which can be written as

$$W \circ P = 0 \ , \qquad (9)$$

where $$W = \begin{vmatrix} \sum (x_i - x_0)^2 & \sum (x_i - x_0)(y_i - y_0) & \sum (x_i - x_0)(z_i - z_0) \\ \sum (x_i - x_0)(y_i - y_0) & \sum (y_i - y_0)^2 & \sum (y_i - y_0)(z_i - z_0) \\ \sum (x_i - x_0)(z_i - z_0) & \sum (y_i - y_0)(z_i - z_0) & \sum (z_i - z_0)^2 \end{vmatrix},$$

and $$P = \begin{vmatrix} a \\ b \\ c \end{vmatrix}$$

We need to solve equation (9) to obtain the least square plane. By imposing

$$a^2 + b^2 + c^2 = 1 \qquad (10)$$

we avoid getting the trivial solution $a = b = c = 0$. With the condition (10), it can be shown that the solution to (9) becomes an Eigenvalue problem, i.e.

$$W \circ E = V \circ E \qquad (11)$$

with $V = |v_1, v_2, v_3|$ are the eigenvalues and $E = |e_1, e_2, e_3|$ the eigenvectors. The eigenvalues are the values of the squared sums of the distances and the eigenvectors are solutions to the *a, b, c* values. We then simply need to choose the set associated with the smallest eigenvalue, corresponding to the smallest squared distance. The set corresponding to the largest eigenvalue will be the worst-fitted plane. All the eigenvectors are perpendicular to each other. If the eigenvalues are similar within a small margin, then the point coordinates are either symmetric or degenerated.

14.2 Pseudo-Code Listings

14.2.1 Pseudo-Code for the CoroViz Tracking algorithm

Listing 2: Main algorithm used for the CoroViz Vessel tracking

```
ALGORITHM TrackCenterTree
[t01]   FOR (all seed points) DO
[t02]      initialize wavefront with seed point
[t03]      WHILE (wavefront NEITHER empty NOR too large) DO
[t04]          UpdateWaveFront
[t05]          SplitWaveFrontIntoRegions
[t06]          FOR (all regions in regionslist) DO
[t07]                FindCenterOfRegion
[t08]               add center to center_line
[t09]          ENDFOR
[t10]      ENDWHILE
[t11]      move all points from center_line to center_tree
[t12]   ENDFOR
[t13]   PruneCenterTree
ENDALGORITHM
```

Listing 3: Algorithm used for updating the wave front

```
ALGORITHM UpdateWaveFront
[u01]   copy wavefront to newfront
[u02]   empty wavefront
[u03]   FOR (all points in newfront) DO
[u04]      find all neighbors
[u05]      FOR (all neighbors) DO
[u06]        IF (neighbor meets inclusion criteria)
                AND (neighbor NOT part of wave)
                AND (neighbor NOT part of newfront) THEN
[u07]           add neighbor to wavefront
[u08]         ENDIF
[u09]      ENDFOR
[u10]   ENDFOR
ENDALGORITHM S
```

The goal of the `UpdateWaveFront` algorithm is to mimic the actual propagation of the blood inside the vessel. The speed function is greater than zero if the global and local inclusion criteria are met ([u06]). In the current implementation, the only inclusion criteria is the seed-point dependent threshold. The neighborhood definition used is a 26-neighborhood.

Listing 4: Algorithm used for splitting up the region growing

```
ALGORITHM SplitWaveFrontIntoRegions
[s01]   create new empty regionlist
[s02]   FOR (all points in wavefront) DO
[s03]      FOR (all regions in regionslist)
[s04]         IF (point connected to region) THEN
[s05]            add point to region
[s06]         ENDIF
[s07]      ENDFOR
[s08]      IF (point not added to existing region) THEN
[s09]         create new region
[s10]         add point to region
[s11]         add region to regionlist
[s12]      ENDIF
[s13]   ENDFOR
ENDALGORITHM
```

The algorithm `SplitWaveFrontIntoRegions` is used to find the different sets of inter-connected voxels in the current wavefront. Again, a 26-neighborhood is used as the connection criteria ([s04]).

Listing 5: Two methods to find the center of a region

```
ALGORITHM FindCenterOfRegion
[f01]   FOR (all points in region) DO
[f02a]     set center to brightest point
[f03]   ENDFOR
ENDALGORITHM

ALGORITHM FindCenterOfRegion (variation)
[f01]   FOR (all points in region) DO
[f02b]     set center to center of mass
```

```
[f03]   ENDFOR
ENDALGORITHM
```

There are two methods currently employed to find the center of a wavefront region: Either the center is set to the point with maximum intensity ([f02a]) or to the average of all point coordinates ([f02b]).

Listing 6: Algorithm to thin out the set of center voxels

```
ALGORITHM PruneCenterTree
[p01]   copy points from center_tree to temporary_tree
[p02]   empty center_tree
[p03]   FOR (all points in temporary_tree) DO
[p04]      find distance to all points in center_tree
[p05]      IF (minimal distance is larger than PRUNEVALUE) THEN
[p06]         add point to center_tree
[p07]      ENDIF
[p08]   ENDFOR
ENDALGORITHM
```

The algorithm `PruneCenterTree` is used to eliminate duplicates and to thin out the set of points found by the region-growing from all seed points.

14.3 Hardware and Software Requirements

Required:

- A 2 GHz (or faster) Pentium PC with at least 1GB memory, running the Windows 2000 / XP operating system.
  (The CoroViz Software was tested and developed on Dell Dimension 8300 Windows XP desktop computer with a single 3.0 GHz Pentium 4 Processor and 1GB memory.)

- 3-button mouse. Alternatively, the scroll wheel button can be used, but has to be explicitly assigned to the middle button in the mouse control panel (depending on manufacturer). Without the middle button default windowing, panning and zooming behavior is not possible!

- While the CoroViz software is distributed to work on the IDL 6.1 virtual machine with no license required, extensive testing of this configuration has not been done. It is recommend to install and run IDL 6.1 with full license, and then compile and load all CoroViz project files.

The Pacman tools must be installed on the scanner to export PAR/REC files. Please refer to the Phillips documentation for further information on file import / export.

14.4 CoroViz User Manual

Whole-heart coronary MRA has been demonstrated to allow the imaging of the entire coronary tree in a single volume. However, a tool for comprehensive display of the 3D data sets has been lacking. To address this need, the CoroViz software was developed.

The main design principle is a restriction to visualization ideas that are transparent to the human operator. All the a-priori knowledge used while creating the visualizations is disclosed. In addition, the different visualization modules are depicted in the same three-dimensional scene, allowing a comparison and assessment of the spatial composition. The scene can be interactively zoomed, rotated, panned, and windowed.

The look-and-feel is made similar to standard software packages such as Gyroview or EasyVison, therefore users with general scanner software experience will feel familiar with the graphical user interface. Further information about the CoroViz project and the proposed visualizations can be found in the accompanying Master's tesis.

The currently implemented modules include:

- Planar - Multi-Planar Reformat (MPR)
- Curved – Curved Reformat
- Globe Viz - Globe Visualization
- Segmentation
- Filtering - Hessian-Matrix based Filtering
- Tube Viz - Masked Volume Projection
- Tracking
- Measurements

14.4.1 Data import

The image data from the scanner has to be exported as a VRREC file pair. These two files consist of a .PAR and a .REC file. The .PAR file, or parameter file, is an ASCII file that contains a subset of the image parameters available through the data base interface. The .REC file, or data file, contains the raw image data. Both files together describe a single series and its images. The prefix represents the version number. Currently, only version 3 is supported.

14.4.2 Graphical User Interface

Figure 52 displays a screenshot of the CoroViz graphical user interface. An axial, coronal and sagittal viewport are permanently shown in the upper three screens.

Whenever there are two or more points defined on the vessel, a cross-sectional viewport is displayed in the lower right screen. The plane normal of the cross-sectional image correspondents to a natural spline fitted through the points.

In the lower left screen, two-dimensional data of the current visualization module is shown whenever available. Options for the various visualization modules can be set in the lower middle screen.

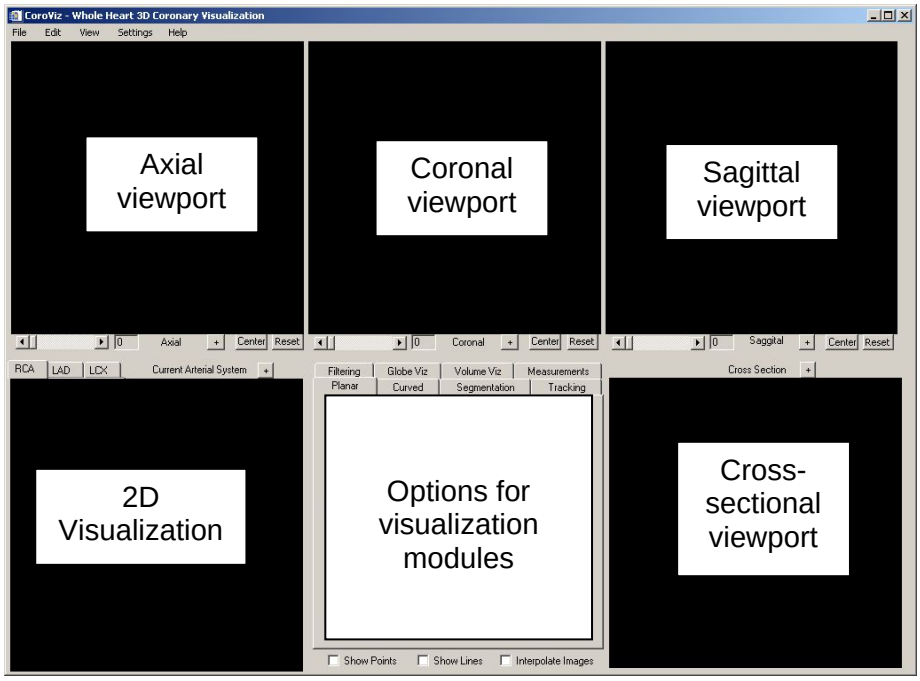


Figure 52: CoroViz Graphical User Interface.

14.4.2.1 *Viewports*

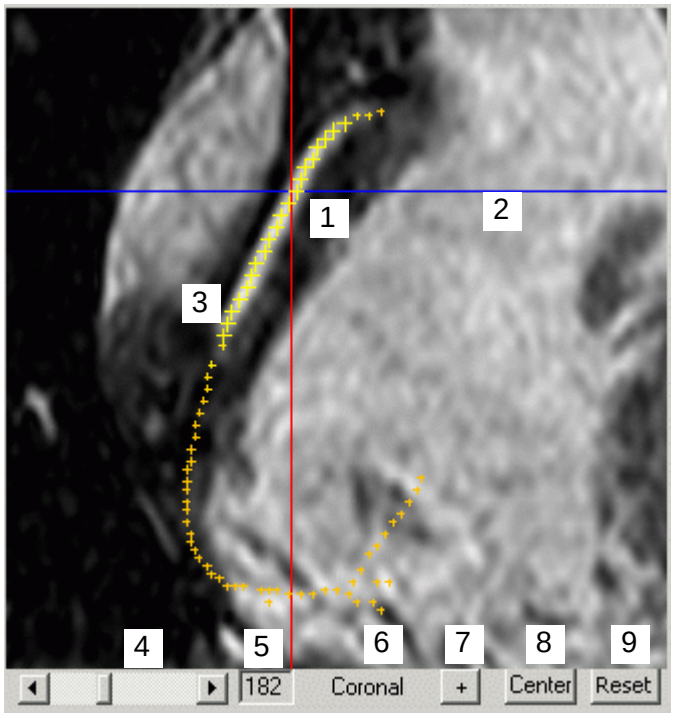


Figure 53: CoroViz viewport.

1. The current point position is marked as the cross-section of the lines.
2. The position of the axial plane is marked with a line.
3. Points close to the view plane are marked brighter and larger than points farther apart.
4. The current view plane position is shown and can be adapted by moving the slider or pressing the arrow keys.
5. Numerical display of the used plane.
6. Name of the viewport (Axial, Coronal, Sagittal, Cross-Section, or 2D Visualization)
7. Clicking this button opens the view port in the Zoom window.
8. Clicking this button centers the viewports at the current point position.
9. Clicking this button opens a menu. Either the windowing, the panning and zooming, or both can be reset to the standard values.

14.4.2.2 *Mouse Control*

In general, it is attempted to keep the mouse button functionality similar to standard software packages such as Gyroview or EasyVison. The mouse can be either clicked or dragged to allow smooth transitions. The mouse button assignments are as follows:

- **Left Button: set current coordinates.**
- **Right Button: add point at current coordinates.**
- **Center Button: control image contrast (window width and center). Delete closest point.**
- **Left-Center Buttons (together): pan the image.**
- **Right-Center Buttons (together): zoom the image.**
- **Right-Left Buttons (together): jump to closest point defined**

14.4.2.3 *Keyboard Control*

The **shift key** is pressed to select and delete points used in the measurement procedure.

The **arrow keys** can be used in all viewports and the 3D visualizer. The function depends on the currently active visualization module. The desired viewport has to be activated by mouse-clicking.

14.4.2.4 *Zoom window*

All the viewports can be opened in a zoom window for closer inspection. User interaction as well as keyboard/mouse control remain identical. The window can be saved as JPG, lossless PNG, or TIFF. In addition, the data from the zoom window can be copied to the clipboard either in bitmap or vector representation for further uses in other programs (e.g. Microsoft Office, Adobe Photoshop). The size of floating Zoom window can be adapted by changing the size of the window frame. Figure 55 shows a screenshot of the zoom window.

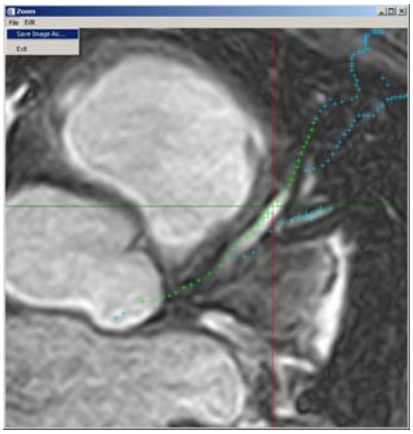


Figure 54: Zoom window.

14.4.2.5 *Menu structure*

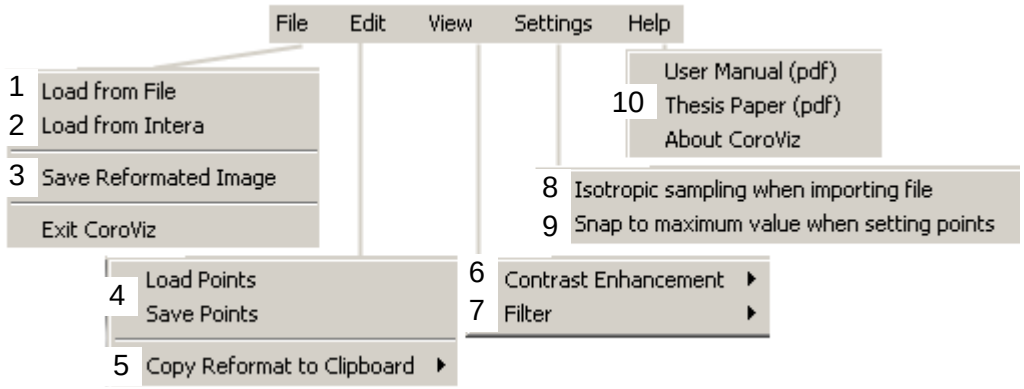


Figure 55: CoroViz menu structure.

1 Open the data set. Data has to be in the PAR/REC format. Currently, only version 3 is supported. Importing and resampling can take up to 40 s depending on the computing power available.

2 Remote loading from the Intera scanner is not yet supported.

3 The reformatted image can be saved as JPG, lossless PNG, or TIFF

4 Points can be saved to disk and loaded on the same or different datasets for comparison.

5 The reformatted image can be copied to the clipboard either in bitmap or vector representation for further uses in other programs (e.g. Microsoft Office, Adobe Photoshop).

6 Contrast enhancement filters include adaptive and non-adaptive histogram equalization. The filter is applied to all two-dimensional images as well as to the texture images of the globe-visualization.

7 Additional filters include *Smooth* (Gaussian smoothing), *Sharpen* (Unsharp masking), *Vessel Enhancement* (two-dimensional vessel filter based on the Hessian matrix), and *Edge Enhancement* (based on the Roberts edge enhancement operator).

8 Setting this option affects data import. Isotropic resampling is needed for the measurement module for accurate results.

9 Setting the snap-to-maximum option will cause manually set points to snap to the brightest point in a 26 Voxel neighborhood of the actual point clicked.

10 User manual and the original Master's thesis are provided as help and description of the visualizations involved.

14.4.3 Workflow

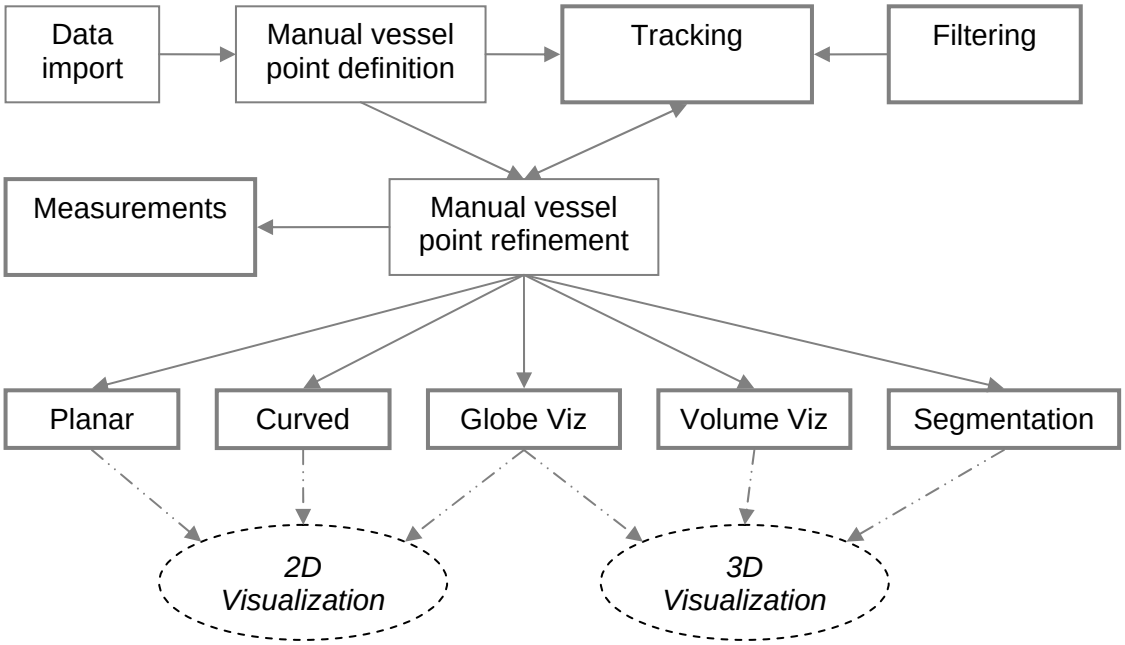


Figure 56: Workflow in the CoroViz program. The modules shown with solid-thick boxes are described in separate sections.

14.4.4 Multiplanar Reformat (MPR)

The aim of the multi-planar reformatting step is (a) to fit a plane through the points defined on the vessel, and (b) to move the plane perpendicular to the plane normal. This step is conceptually similar to rotating and moving the volume so the vessel is centered and lies in plane with the volume slices.

The minimum number of points needed to define a plane is three. If there are more points defined on the vessel, the plane is fitted in a way that the sum of the squared distances from the points to the plane is minimized. Figure 57 shows the available options.

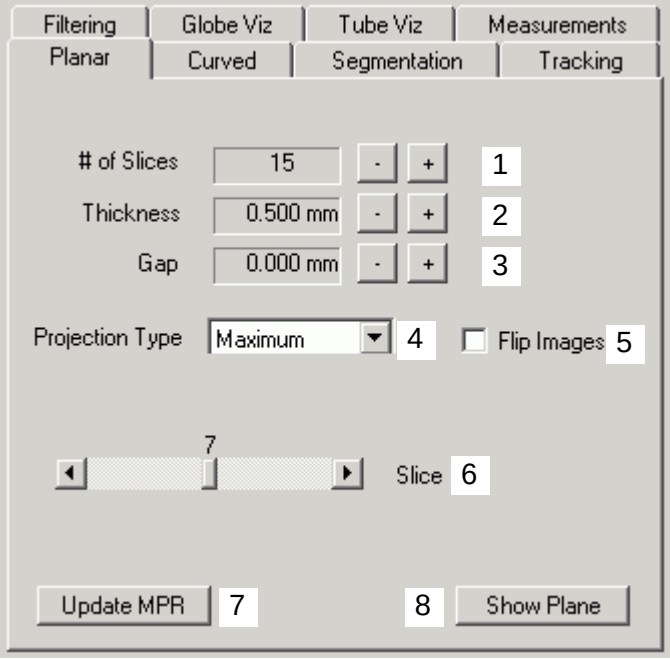


Figure 57: Options for multi-planar reformat visualization.

1 The number of slices in the reformatted image stack. The upper limit is the memory available.

2 The slice thickness defines the width of the projection volume in direction of the plane normal.

3 The distance between adjacent projection volumes is defined as the slice gap, with a negative gap indicating an overlap of the projection volumes

4 Types of projections include maximum (MIP), minimum (MinIP) and average intensity.

5 Numerical display of the used plane.

6 Currently shown slice in the lower left viewport. Can be adapted by either clicking on the slider or pressing the arrows keys when the 2D viewport is activated.

7 Clicking this button starts the computing of the MPR image stack.

8 Clicking this button starts the 3D Visualization tool and displays the plane in the 3D scene.

14.4.5 Curved Reformat

The goal of curved reformation is to make a vessel visible in its entire length in one image. First we define a surface for the wanted area that is both smooth and intersects the points. A Delaunay triangulation of a planar set of points is created, only taking the X- and Y-coordinates of the points into account. Once the triangles are defined, the Z-coordinates of the points are taken back into account and the triangles are hung in a 3D space, creating a two-dimensional surface. Once all the triangles are defined, the Z-coordinates are interpolated for the desired grid coordinates.

The interpolated points define a surface through the reformatted volume. The data is then sampled at the surface positions using trilinear interpolation. The current implementation simply projects the sampled values onto the x, y plane. Figure 58 shows the available options.

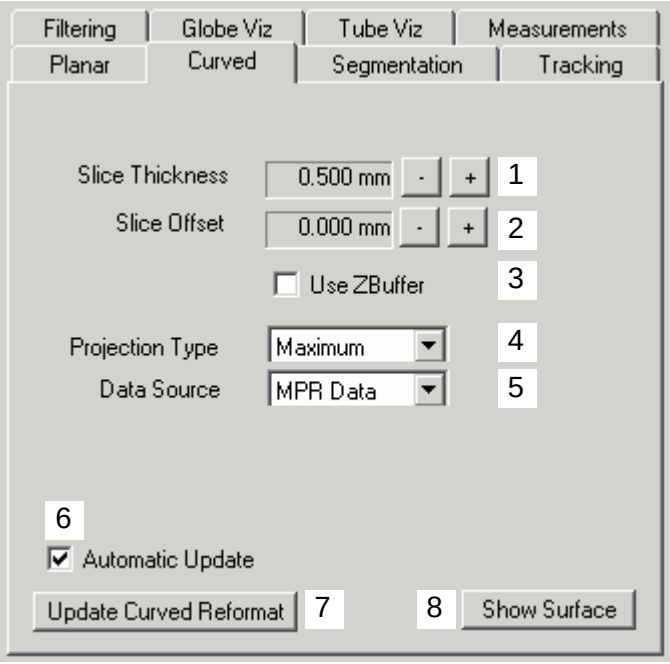


Figure 58: Options for the curved reformat visualization.

1 The slice thickness defines the width of the projection volume in direction of the elevation axis of the reformatted volume.

2 The slice offset n (may be negative!) is added to the elevation coordinates prior to the sampling.

3 If the Z-Buffer option is set, new points defined on the curved reformat are set to the maximum (for MIP) or minimum (for MinIP) point used in the projection of the curved reformat. Setting the Z-Buffer options has no effect if the the type of projection is set to average.

4 Types of projections include maximum (MIP), minimum (MinIP) and average intensity.

5 The curved reformat can be done with either the original data or the MPR image stack as data source.

6 Setting the Automatic Update option causes the curved reformat image to be automatically

updated whenever points are defined / deleted or parameters have changed.

7 Clicking this button generates the curved reformat image

8 Clicking this button starts the 3D Visualization tool and displays the curved surface in the 3D scene.

14.4.6 Vessel Segmentation

The CoroViz segmentation is a simple vessel lumen segmentation based on a three-dimensional variation of the full with at XX % maximum criterion (FWXX%M). Basically, all voxels surrounding the center points that have at least an intensity of XX % are considered to be part of the vessel. To prevent a leaking out of the segmentation, the distance of the included voxel is limited to the maximum expected vessel radius. Figure 59 shows the available options.

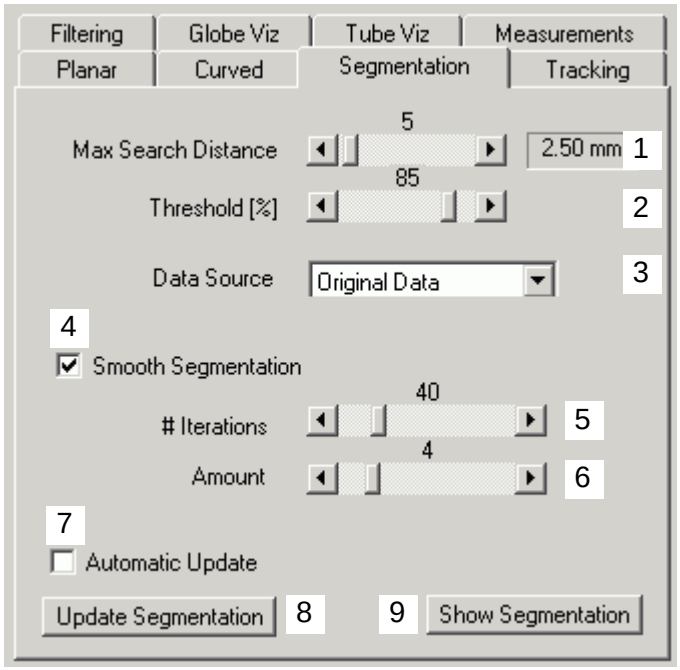


Figure 59: Options for the vessel segmentation visualization.

1 The maximum search distance restricts the inclusion of voxels far away from the seed points and prevents leaking out of the segmentation.

2 The percentage value for the vessel border criterion.

3 For demonstration purposes the segmentation can also be done based on the vessel-enhanced filtered data.

4 Setting the Smooth Segmentation option performs Laplacian smoothing on the polygon mesh output as described by the following formula:

$$\vec{x}_{i,n+1} = \vec{x}_{i,n} + \frac{\lambda}{m}\sum_{j=0}^{m}\left(\vec{x}_{j,n} - \vec{x}_{i,n}\right)$$

where $\vec{x}_{i,n}$ is the vertex *i* for iteration *n*, λ is the smoothing factor, and *m* is the number of vertices that share a common edge with $\vec{x}_{i,n}$.

5 The number of smoothing iterations (see above).

6 The smoothing factor applied (see above).

7 Setting the Automatic Update option causes the segmentation to be automatically updated whenever points are defined / deleted or parameters have changed.

8 Clicking this button generates the segmentation

9 Clicking this button starts the 3D Visualization tool and displays the vessel segmentation in the 3D scene.

14.4.7 Hessian-based Vessel Filtering

Vessel enhancement filtering based on all eigenvalues of the Hessian matrix was first proposed by (Frangi, Niessen et al. 1998). The idea of the vesselness filter is to enhance voxels which have a high probability of being part of a vessel. To locally determine that likelihood, an eigenvalues analysis on the Hessian matrix of the Gaussian-filtered data is performed. For the further discussion, the eigenvalues λ_i shall be ordered in increasing magnitude ($|\lambda_1| \le |\lambda_2| \le |\lambda_3|$)

The first ratio is defined as $R_A = \frac{|\lambda_2|}{|\lambda_3|}$ and is able to distinguish between tubular-like ($R_A \approx 1$) and plate-like structures ($R_A << 1$). The second ratio $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \cdot \lambda_3|}}$ is a measure for the deviation from a blob-like structure. R_B is maximal for a spherical structure, but cannot differentiate between a plane-like and a tubular-like structure.

We use the structure parameter $S = \sqrt{\sum_{i \le 3} \lambda_i^{\,2}}$ which is small in the background to suppress filter response in background regions. The three geometric parameters above are combined to a vessel enhancement filter together to

$$V(\vec{x}, w) = \begin{cases} 0, & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ v(\vec{x}, w), & \text{otherwise} \end{cases}$$

where $v(\vec{x}, w) = (1 - e^{-(R_A^{\,2}/2\alpha^2)}) \cdot (e^{-(R_B^{\,2}/2\beta^2)}) \cdot (1 - e^{-(S^2/2\gamma^2)})$

The parameters α, β, and γ adapt the filter response to the distinction between plate- and tubular-like structures (α, R_A), deviation from blob-like structures (β, R_B), and magnitude of background suppression (γ, S). Figure 60 shows the available options for the Hessian-based filtering.

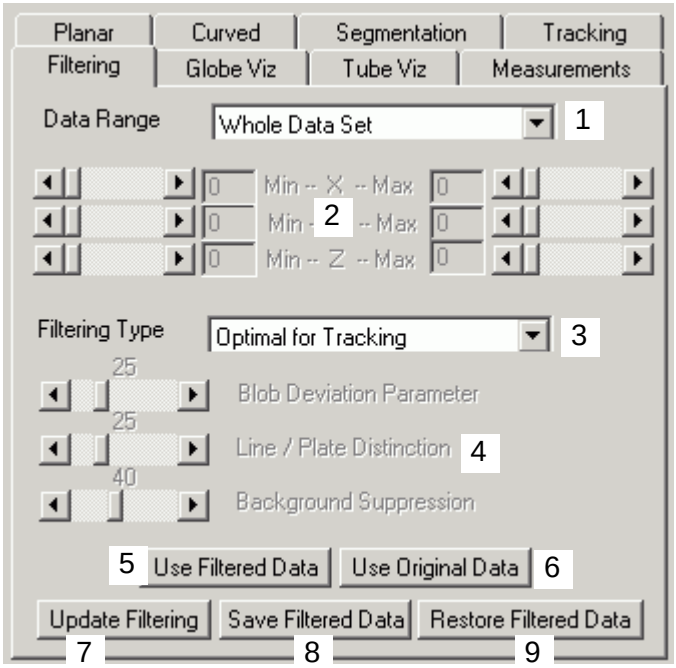


Figure 60: Options for Hessian-based filtering.

1 The filtering process can be restricted to a subset of the original data. Options include a manual delimitation or the restriction to a volume that encompasses the points on the vessel defined so far.

2 The filter volume can be defined here if the Data Range option is set to User defined.

3 The parameters α, β, and γ (see above) are either set manually, or default values for an optimal tracking response can be used.

4 If Filtering type is set to Full parameter control, the parameters α, β, and γ can be manually adjusted.

5 The CoroViz software allows the use of the filtered data instead of the original data for all visualization modules.

6 Clicking the Use Original Data button resets the used data for all modules back to the original data.

7 Clicking this button starts the filtering process. **Warning: Depending on data range and computer hardware this process might take several minutes.**

8 The vessel-enhanced filtered data can be saved to and restored from file to facilitate frequent processing of the same dataset. Clicking this button saves the data to disk.

9 Checks if filtered data is available for the current dataset and restores the filtered data from disk.

14.4.8 Vessel Tracking

The CoroViz tracking algorithm implements a wavefront and region-growing approach. Main advantages are:

- Inherent branch detection capability
- Wave propagation process is closely related to the actual physical blood flow inside the vessel (‘where the blood flows, the wave will pass as well’)
- The operator can easily add new seed points or manually correct the found center-line.

The current implementation uses the vessel enhancement filter as a preprocessing step. The tracking is carried out separately for the RCA, the LAD, and the LCX according to the following algorithm:

```
ALGORITHM TrackCenterTree
[t01]   FOR (all seed points) DO
[t02]      initialize wavefront with seed point
[t03]      WHILE (wavefront NEITHER empty NOR too large) DO
[t04]          UpdateWaveFront
[t05]          SplitWaveFrontIntoRegions
[t06]          FOR (all regions in regionslist) DO
[t07]                FindCenterOfRegion
[t08]               add center to center_line
[t09]          ENDFOR
[t10]      ENDWHILE
[t11]      move all points from center_line to center_tree
[t12]   ENDFOR
[t13]   PruneCenterTree
ENDALGORITHM
```

The region-growing process is started from each seed point ([t01]). The wavefront propagates until no additional voxels can be found. Also, if the wavefront explodes (e.g. due to a low threshold), the propagation is stopped ([t03]) to prevent exhaustive computations. The current wavefront is split into different regions ([t05]), and the center voxel is found for each region separately ([t07]). The new center voxels are then added to the previously found center line [t08].

Since the region-growing process is started from each point separately, the resulting coronary tree will have duplicate branches, especially if the seed points are close together and the search regions as defined by the maximum search distance overlap. A pruning of the tree is therefore necessary ([t13]). Figure 61 shows the available options for the vessel tracking step.

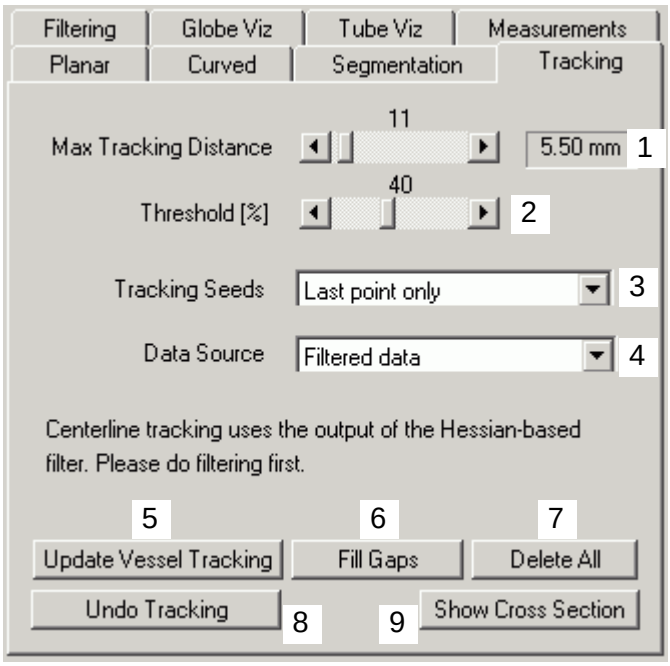


Figure 61: Options for the vessel tracking process.

1 The maximal search distance is used to limit the search area to a cubic volume around the used seed points.

2 The threshold value determines how bright a voxel has to be (in relation to the value at the seed point) in order to be considered part of the vessel structure.

3 Iterative tracking is possible by using points on the found center line as the new seed points for the next wave propagation. Since the threshold value depends on the value at the given seed point position, the search process adapts to the local brightness distribution.

The use of all points - together with a small maximum search distance to limit computational expense – is very useful in detecting additional side branches.

A limitation to the end points works well for the distal elongation of vessel branches.

The last-point-only mode is used frequently if the operator defines one additional point on the coronary vessel and would like to start a new tracking from there.

4 For testing purposes, the tracking may be done based on the original data as well as the filtered data.

5 Clicking this button performs the tracking with the given options.

6 Segmentation and tube visualization depend on a connected vessel center-line. Clicking this button tries to fill the gaps.

7 Clicking this button deletes all points on the current vessel.

8 Clicking this button resets the current vessel points to the set defined before the last tracking was performed.

9 Clicking this button starts the 3D Visualization tool and displays the current cross-section image in the 3D scene.

14.4.9 Globe Visualization

The aim of the globe visualization is to deform a sphere so that all points belonging to the coronal tree lie on its surface. Surface creation and globe projection are very similar to the concepts employed for the curved reformat. However, instead of bending a plane, we deform a sphere, and instead of projecting in z-direction, we calculate the MIP in radial direction.

The points $P_i' = \{lon_i, lat_i, r_i\}$ can be compared to local measurements of longitude lon_i, latitude lat_i, and elevation r_i. The object is to find a surface that is both smooth and intersects the points $\{lon_i, lat_i\}$ at elevation r_i. The Delauney triangulation is created by connecting all generating points which share a common tile edge. Once all the triangles are defined, the elevation coordinates r_i may be interpolated for the desired grid coordinates $\{longrid, latgrid\}$ using e.g. *linear*, *quintic*, *minimum curvature*, *kriging* or *thin-plate spline interpolation*. The values of $r\ (longrid, latgrid)$ depicts the elevation as a function of the longitude and latitude coordinates of the grid array. The interpolated points $\{longrid, latgrid, r(longrid, latgrid)\}$ define a surface through the original data volume. First, the interpolated points are transformed back into the Cartesian coordinate system. The data is then sampled at the surface positions using trilinear interpolation.

In continuing analogy to geodesy, the sampled surface would be a map of the coronaries. We can then wrap the map around the three-dimensional globe model, a process called *texture mapping*. Figure 62 shows the analogy between covering the earth globe with a map of the continents and warping the surface image to the coronary globe. Figure 62 shows the available options for the globe visualization.

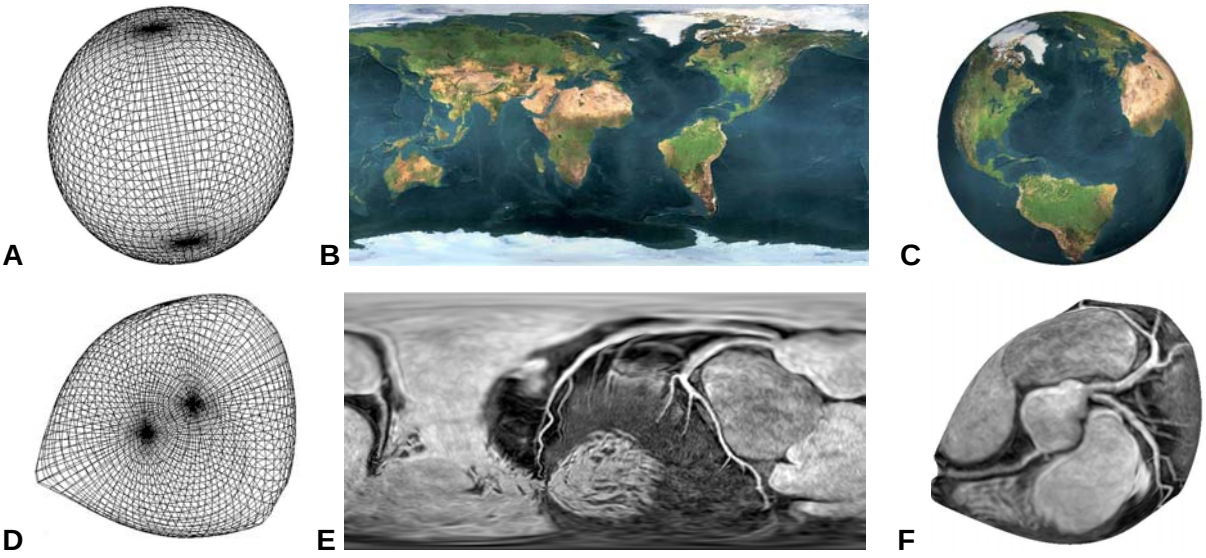


Figure 62: Analogy between GlobeViz texture mapping and earth map projection. The earth map (B) is projected onto a sphere mesh (A), resulting in a three-dimensional model of the earth globe (C). Similarly, a map of the coronaries (E) can be warped onto the interpolated surface grid (E). The resulting GlobeViz model is also three-dimensional and preserves length and area.

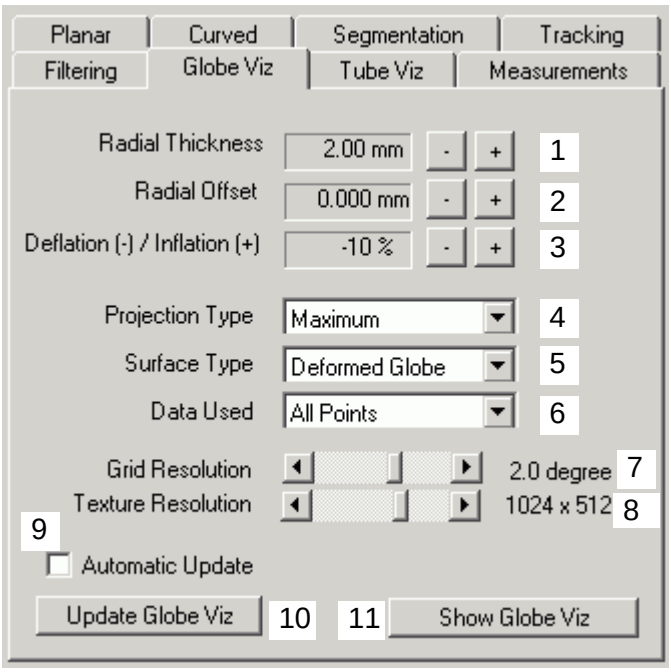


Figure 63: Options for the GlobeViz visualization module.

1 The thickness defines the width of the projection volume in radial direction. By increasing the radial thickness of the GlobeViz visualization, is visible even if the defined center-line is incomplete.

2 The radial offset allows the operator to grow and shrink the surface of the globe in radial direction (i.e. from and towards the center point).

3 The deflation / inflation options restricts the growing process to areas away from the coronaries. Growing and shrinking is similar to inflating and deflating a balloon where the defined vessel points act as a restricting force.

4 Types of projections include maximum (MIP), minimum (MinIP) and average intensity.

5 Instead of mapping the interpolated texture image on the deformed globe, it's also possible to project the image onto the initial sphere. **Warning: true distances in the three-dimensional model are no longer preserved.**

6 It may be useful to limit the set of used points to a sub-set of all vessels used for globe visualization procedure. Options include the use of points from the current vessel only or for any combination of RCA, LAD, and LCX.

7 The grid resolution defines the mesh (longitude and latitude) used in creating the surface. If the grid resolution is too low, important vessel features might not be depicted because the resulting sampling surface does not pass through all defined vessel points.

8 The creation of the smooth surface grid is an expensive operation. The three-dimensional grid positions are linearly interpolated in order to sample at points in between the grid defined by the surface, resulting in a higher texture image resolution.

9 Setting the Automatic Update option causes the globe visualization to be automatically updated whenever points are defined / deleted or parameters have changed.

10 Clicking this button generates the globe visualization.

11 Clicking this button starts the 3D Visualization tool and displays the globe visualization in the 3D scene.

14.4.10 Tube Visualization

Maximum intensity projections are still considered the gold-standard in viewing MRA data. For coronary MRA as provided by the whole-heart sequence, MIP over slices or the whole data set do not work well because the arteries are over-projected by brighter organs surrounding the coronaries. We therefore introduce a module called ‘TubeViz’, which is based on masked volume projection. The mask consists of a tube with user-defined diameter along the vessels. The volume rendering function is again chosen as a maximum intensity projection: the value of each pixel on the viewing plane is set to the brightest voxel, as determined by its opacity. Figure 64 displays the mask as well as the resulting projection. The available options for the tube visualization are shown in Figure 64.

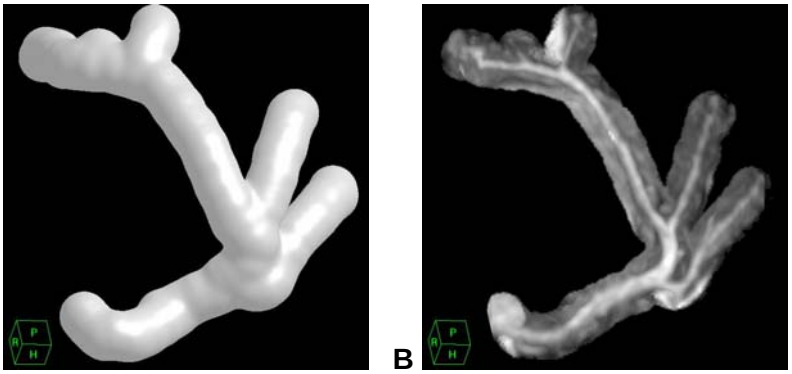


Figure 64: Masked volume projection ('TubeViz') of the left circumflex. The mask consists of a tube with user-defined diameter along the vessel (A). The tubular maximum intesity projection is performed with the original data (B), allowing an optimal assessment of the vessel structure without over-projection from other organs.

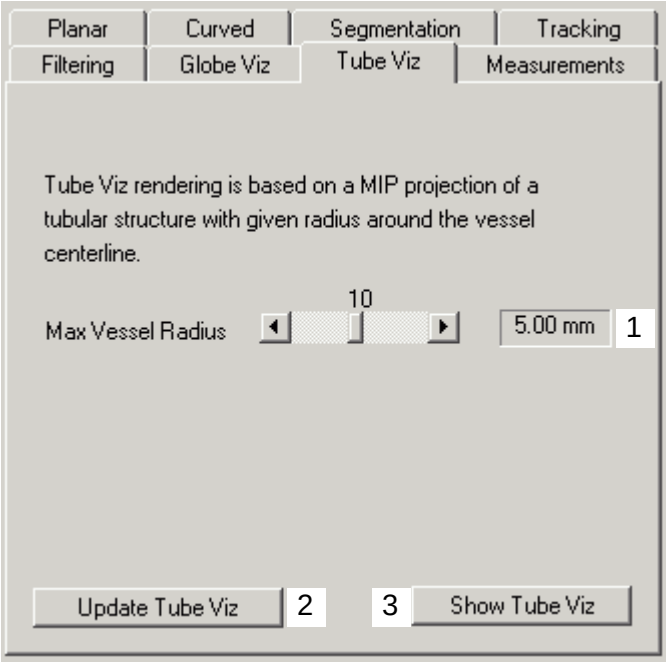


Figure 65: Options for the tube visualization.

1 The tube radius is the only parameter input for the TubeViz visualization. Over-projection from other organs is minimized by choosing a small radius. On the other side, a large radius allows the visualization of branching vessel parts even if they have not been marked with vessel points. Therefore, the tube radius is usually a trade-off between coverage and visibility.

2 Clicking this button generates the tube visualization.

3 Clicking this button starts the 3D Visualization tool and displays the tube visualization in the 3D scene.

14.4.11 Measurement and Quantification

Length measurement is based on a user-defined selection of vessel points. The operator first highlights points for the wanted vessel length and then presses the update button. **Selection of points for measurement is done by pressing the shift key and clicking on points or dragging the right mouse button in all viewports and in the 3D Visualizer. To delete points from the selection, press the shift key and click or drag the middle mouse button.**

The length of a natural spline fitted through the points is taken as the approximation of the vessel length. The user has full control over the points used in defining the spline, and visual three-dimensional feedback on the distance measured is provided.

The implemented diameter and area quantification module relies on the knowledge of the vessel centerline as defined by the tracking algorithm. In a first step, vessel lumen segmentation on the cross-sectional image is obtained at the point of investigation, using an algorithm similar to the one used in the segmentation module. Instead of relying on a fixed percentage value for the *full with at % maximum criterion*, the operator can adapt the percentage to a value that provides adequate lumen segmentation for the image modality and method used. Secondly, a radial di-

ameter sweep is performed from the brightest point in the cross-sectional image to the detected borderline. The minimum, maximum, and average diameter is then recorded and the quantification procedure repeated for each user-defined vessel point. The measurements are shown in the lower right viewport. Figure 66 shows the measurements option panel.

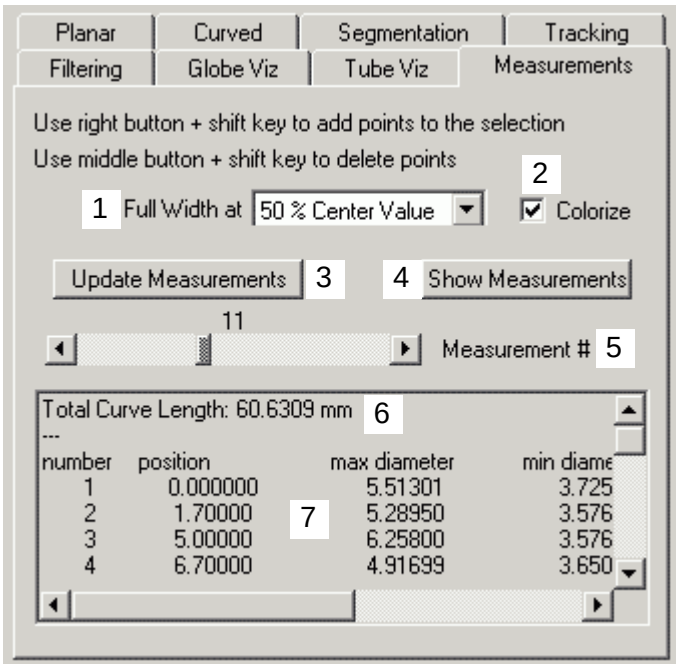


Figure 66: Options and output for the measurement module.

1 The percentage value for the vessel border criterion (is chosen depending on the data acquiring method).

2 Setting this option highlights the found vessel lumen in the measurement control images.

3 Clicking this button updates the measurement process.

4 Clicking this button starts the 3D Visualization tool and displays the performed measurements in the 3D scene.

5 The current measurement control image can be adapted by either clicking on the slider or pressing the arrow keys when the lower left viewport is activated. The current point in all the other viewports is automatically set to the center of the current measurement.

6 The total length of the spline fit through the selected measurement points.
Warning: inclusion of side branches will cause the spline to loop back and corrupts the length measurement. Please check the detected spline in the 3D Visualizer.

7 The obtained measurement data can be copied (press CTRL – C) and pasted into Excel or any other spreadsheet application for further processing.
Warning: diameter and area measurements are frequently corrupt due to nearby bright structures and should first be investigated using the measurement control images.

14.4.12 3D Visualizer

Clicking on any of the Show [MODULE] buttons opens the floating 3D Visualizer. **Even though the 3D Viewport is active, the CoroViz user interface remains working. Updating of visualization modules, setting and deleting points, and parameter changes are all automatically updated in the 3D Visualizer.**

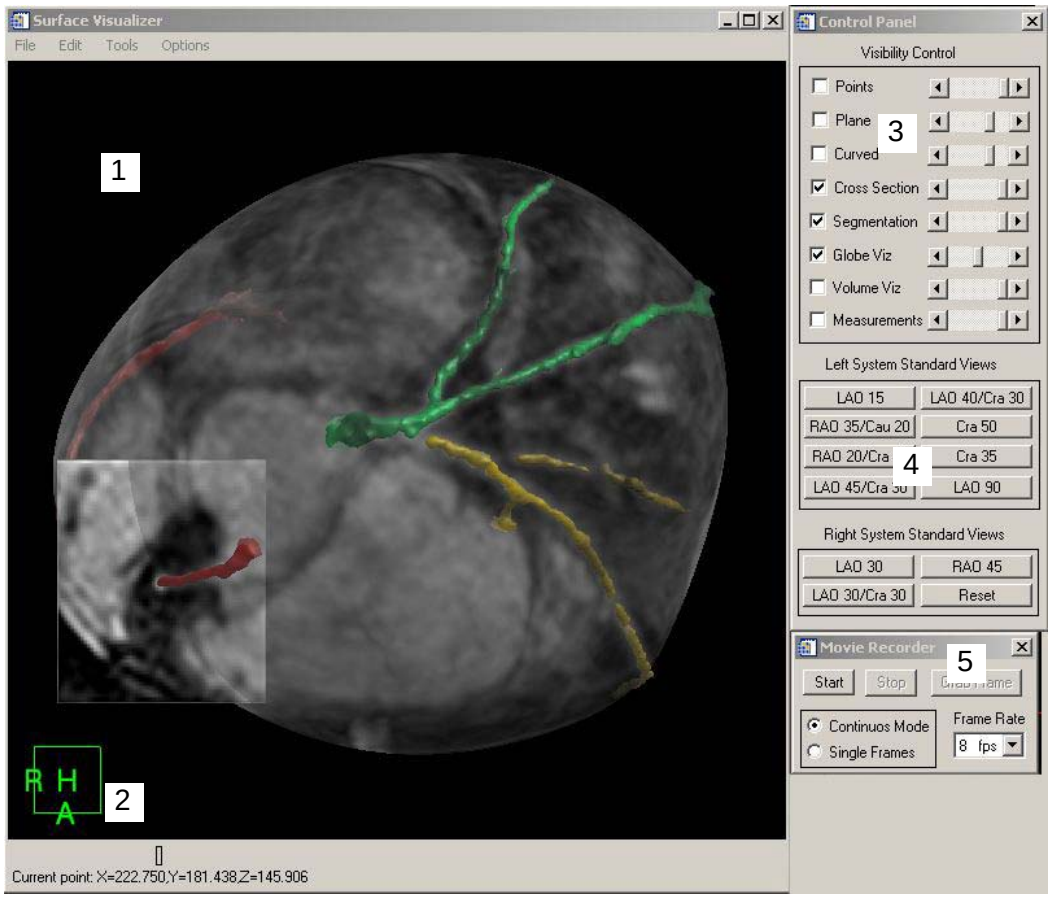


Figure 67: CoroViz 3D Visualizer

1 The three-dimensional viewport can be interactively rotated, panned, zoomed and windowed. The mouse button assignments are as follows:

- Left Button: set current coordinates.
- Right Button: add point at current coordinates.
- Center Button: control image contrast (window width and center). Delete closest point.
- Left-Center Buttons (together): pan (move) viewport
- Right-Center Buttons (together): zoom viewport
- **Right-Left Buttons (together): rotate viewport**

2 An orientation cube indicates the current view direction. A=Anterior, P=Posterior, R=Right, L=Left, H=Head, F=Feet.

3 Visibility and transparency for all visualization modules can be set in the visibility control panel.

4 8 standard views for the left system and 3 standard views for the right system provide a fast overview of the coronary tree.

5 An intuitive movie recorder allows the fast generation of animations (see below)

14.4.13 Movie Recorder

The movie recorder is based on a WYSIWYG (what you see is what you get) idea. In the *continuous mode*, a frame from the 3D viewport is grabbed whenever anything changes. The operator navigates through the data, blends in the different visualization modules - and all is captured automagically in the animation. In the *single frames mode*, a frame is added to the animation whenever the user clicks the Grab Frame button. A change between the two modes is always possible. The *single frames mode* is mostly used when the rendering time is too large for smooth transitions, e.g. when using the tube visualization module with its expensive volume projection. For video compression, all video codecs that are currently installed on the machine can be used. For optimal compatibility (also with Microsoft Powerpoint) the Cinepak Code by Radius is recommended. The movie recorder uses the excellent IDL2AVI library written by Ronn Kling.

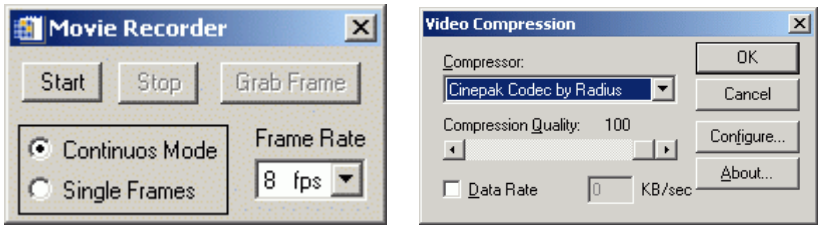


Figure 68: Options for the Movie Recorder module.